November 2016

Applications of Sampling and Estimation on Networks

Fabricio Murai Ferreira
University of Massachusetts Amherst

APPLICATIONS OF SAMPLING AND ESTIMATION ON NETWORKS

A Dissertation Presented

by

FABRICIO MURAI FERREIRA

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

September 2016

College of Information and Computer Sciences

APPLICATIONS OF SAMPLING AND ESTIMATION ON NETWORKS

A Dissertation Presented

by

FABRICIO MURAI FERREIRA

Approved as to style and content by:

_____

Don Towsley, Chair

_____

Krista Gile, Member

_____

David Jensen, Member

_____

Ben Marlin, Member

_____

Bruno Ribeiro, Member

_____

James Allan, Chair of the Faculty
College of Information and Computer Sciences

DEDICATION

To Gabriela and Nicolas.

ACKNOWLEDGMENTS

I thank my advisor, Don, for all he has taught me during this five years. Among other things, he has taught me how to identify and conduct good research; that it is possible to be very busy and yet make yourself available (although I don't expect to master this technique within the next decade). I also thank him for giving me freedom to choose what to work on and for providing me the guidance that I needed. To my wife and son, my deepest appreciation for enduring the toughest times and sharing the most delightful moments of my life. I thank Bruno (Ribeiro), from whom I learned a lot, for his enthusiasm and for the great experience working together. I thank all the people with whom I had the pleasant experience to collaborate during my PhD, including Krista Gile, Gisele L. Pappa and Diogo Rennó. My lab mates and friends (Anand, Bo, Bruce, Chang, Gayane, James, Kun, Mostafa, Roman, Sookhyun, Yeon-Sup, Yung-Chih) for the helpful discussions and fun moments necessary to go through times when stress is elevated. To Cibele and Fernando, for their friendship and for taking care of Nick and Gabi, when I could not make myself present. To Ana, Bruno (da Silva), Marcia, Mariana, Rafael, Raphael, Shiri, Thiago and all the dear friends who were always there for me. To the friends I made during grad school. To my friends and professors from Laboratorio LAND at UFRJ (many of which are somewhere else now). To my mother and friends from Brazil who visited me (Rafael Pinto, Debora, Kikuchi, Talita). To those who did not visit me but always wished me well, including Flavio, Paulo, Rafael, Yanko and Sete. Last, to CNPq-Brazil for funding me during four years.

ABSTRACT

APPLICATIONS OF SAMPLING AND ESTIMATION ON NETWORKS

SEPTEMBER 2016

FABRICIO MURAI FERREIRA

B.Sc., FEDERAL UNIVERSITY OF RIO DE JANEIRO

M.S., FEDERAL UNIVERSITY OF RIO DE JANEIRO

M.S., UNIVERSITY OF MASSACHUSETTS AMHERST

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Don Towsley

Networks or graphs are fundamental abstractions that allow us to study many important real systems, such as the Web, social networks and scientific collaboration. It is impossible to completely understand these systems and answer fundamental questions related to them without considering the way their components are connected, i.e., their topology. However, topology is not the only relevant aspect of networks. Nodes often have information associated with them, which can be regarded as node attributes or labels. An important problem is then how to characterize a network w.r.t. topology and node label distributions. Another important problem is how to design efficient algorithms to accomplish tasks on networks. Since nodes often have attributes, an interesting avenue for investigation consists in learning and exploiting existing correlations between node and neighbor attributes

for accomplishing a task more efficiently. One of the challenges faced when studying networks in the wild is the fact that in general their topology and information associated with its nodes cannot be directly obtained. Thus, one must resort to collecting the data, but when obtaining the entire network is infeasible, sampling and estimation are the best option. This dissertation investigates the use of sampling and estimation to characterize networks and to accomplish a particular task. More precisely, we study (i) the problem of characterizing directed and undirected networks through random walk-based sampling, (ii) the problem of estimating the set-size distribution from an information-theoretic standpoint, which has application to characterizing the in-degree distribution in large graphs, and (iii) the problem of searching networks to find nodes that exhibit a specific trait while subject to a sampling budget by learning a model from node attributes and structural properties, which has application to recruiting in social networks.

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

Networks or graphs[1] are fundamental abstractions that allow us to study many important real systems, such as the Web, social networks, and scientific collaboration. It is impossible to completely understand these systems and answer fundamental questions related to them without considering the way their components are connected, i.e., their topologies. Topology determines, among other things, the speed at which a process takes place on a network. For example, information will propagate slowly on a line graph, but will propagate much faster on a complete graph [41]. In barbell graphs (i.e., graphs formed by connecting two copies of a complete graph by a bridge), it may take a long time for nodes to reach global consensus when they update the value of local variables based on messages exchanged with their neighbors [40]. Topology also determines the robustness of a network against attacks. When nodes are removed uniformly at random, one by one, a graph generated by the preferential attachment model is known to remain connected for a large number of removals [3]. However, if nodes are removed in decreasing order of their degrees, this type of graph quickly breaks into disconnected pieces.

However, topology is not the only relevant aspect of a network. Nodes often have information associated with them. For instance, in a social network, nodes represent individuals whose profiles contain information about places where they have lived, studied, their interests, etc. This information can be regarded as node attributes or labels. An important problem is then how to characterize a network w.r.t. topology and node label distributions.

[1] We use the term *graph* when referring specifically to topology. The term *network* refers to the combination of topology and information associated with nodes and edges.

Another important problem is how to design efficient algorithms to accomplish tasks on networks. One example of such a task is: find researchers that have been cited between 1,000 and 2,000 times in the last five years. Since nodes often have attributes, an interesting avenue for investigation consists in learning and exploiting existing dependencies between node and neighbor attributes for accomplishing a task more efficiently.

One of the challenges faced when studying a network in the wild is the fact that in general its topology and node labels cannot be directly obtained. Some networks, such as the Internet, are composed of several autonomous parts controlled by different institutions that do not have information about each other. Others are controlled by single organizations but their data is just not made available for download. In both cases, one must resort to collecting the data, but when obtaining the entire network is infeasible, sampling and estimation are the best option.

The first part of this dissertation focuses on the use of sampling and estimation to characterize networks w.r.t. node label distributions, i.e. to estimate the fraction of nodes in a network that have a given label. Labels can be any type of information associated with nodes, either topological (e.g., out-degree) or not (e.g., citizenship of an individual). For networks, uniform vertex sampling (VS) and random walks (RW) are two common ways of sampling a network. VS is typically performed by randomly generating a vertex id and then querying this id to see if they correspond to an existing vertex. When the set of valid vertex ids is small relatively to the id space size, RWs are much cheaper than VS.

A random walk on an undirected network can be modeled as a stochastic process that has well known statistical properties. In particular, when the walk reaches steady state (provided it exists), it samples edges equiprobably. A random walk can be used for sampling vertices by taking one endpoint of the sampled edges. In this case, the steady state probability of sampling a node is proportional to the node degree, allowing for simple bias removal. The same property holds for directed networks where both incoming and outgoing edges from a node are observable when the walker navigates the network as if it were

undirected, i.e., ignoring edges' directions. However, in some directed networks (e.g., web graphs), an edge from a to b can be seen from a, but not from b. In this case, the steady state distribution of the classic random walk depends on the entire structure of the network, which renders the method unsuited for characterization.

Some recent works propose RW methods designed for directed networks with invisible in-edges. In this context, we propose Directed Unbiased Frontier Sampling (DUFS), a sampling method based on multiple coordinated RWs, which generalizes two important RW methods, one designed for undirected networks and the other for directed networks with invisible in-edges. We also propose an estimator for node label distribution that can leverage information from the initial walker locations, thus obtaining significant gains in accuracy when the number of walkers is large w.r.t. the total sampling budget.

When in-edges are not directly observable, estimating the in-degree distribution becomes more complicated: there is an extra degree of uncertainty as we only observe some of the incoming edges to a node, i.e., the true in-degree of a sampled node is unknown. Assuming uniform edge sampling as an ideal case for RW sampling, we pose the following question: is it possible to characterize large networks using random walks? Given a fixed sampling probability, what happens to the estimation error when the network grows? We study this problem analytically by using a more general formulation of the in-degree estimation problem with unobservable incoming edges called the problem of estimating the set-size distribution. In this problem, elements are randomly sampled from a collection of non-overlapping sets and we seek to recover the original set size distribution from the samples. Recoverability of the original set size distribution presents a sharp threshold with respect to the fraction of elements that remain in the sets. If this fraction lies below the threshold, typically half of the elements in power-law and heavier-than-exponential-tailed distributions, then the original set size distribution is unrecoverable. These results imply that, when estimating in-degree distributions via uniform edge sampling, the fraction of edges sampled must be above the same threshold in order to obtain accurate estimates.

Last, we consider a problem where sampling and estimation are not used to characterize a network, but rather to accomplish a task on a network. More precisely, we study selective harvesting, the problem of finding as many nodes that exhibit a specific trait as possible, subject to a sampling budget. The network is assumed unknown but as new nodes are sampled we are allowed access to their neighbors and our knowledge of the network grows. This knowledge can be used to learn a model to help us select new nodes to sample. This problem poses several research questions: How to encode the relationship between candidate nodes and the observed network? How to learn a model for estimating labels of a set of nodes that is constantly changing? Unlike most problems in learning, a model must be fit to the observations collected by the same model. Since the model is constantly changing and the network is only partially observed, it is impossible to completely remove the bias from the search.

We design highly informative features that blend node attributes and network structure, and evaluate the performance of several models using those features in finding target nodes. We show that fitting a model to the observations it collects can severely erode its performance, a phenomenon we call the *tunnel vision effect*. We propose a framework called Directed Diversity Dynamic Thompson Sampling (D^3TS), which combines several methods to mitigate the tunnel vision effect by increasing diversity in the training data. This problem has application to recruiting users of social networks for a campaign, when the trait being modeled is the user interest in joining it.

1.1 Contributions

This dissertation provides contributions to the understanding and solving of problems that arise in networks, through the use of sampling and estimation techniques. Our contributions are divided into three parts.

1.1.1 Characterizing node populations

1. We propose the Directed Unbiased Frontier Sampling (DUFS), a method that generalizes previous methods based on random walks. DUFS uses multiple coordinated walkers and random jumps to estimate statistical properties of undirected networks and directed networks regardless of the in-edges visibility. Walkers are initially located on vertices chosen uniformly at random. The probability of performing a jump (instead of moving to a neighbor) is proportional to the degree of the node where a walker is located.

2. We introduce a new estimator of vertex label distributions which combines observations from the initial walker locations with those made during the walks in order to produce better estimates.

3. We investigate the impact of the number of walkers and the impact of the random jump weight – which controls jump probability – on estimation errors associated with DUFS. This allows us to provide practical guidelines on how to set them under different scenarios.

4. We show that DUFS yields smaller errors than other random walk-based methods when estimating probability masses associated with low in-degree and low out-degree values, without hurting estimation performance in the distribution tail.

1.1.2 Characterizing the in-degree distribution

1. We study the problem of estimating the in-degree distribution when incoming edges are not directly observable from a node. Assuming edges are observed via independent edge sampling, we formulate this problem as the – more general – set size distribution estimation problem.

2. We quantify the Fisher information contained in a sample and use Cramér-Rao bounds to derive lower bounds on the estimation error of the set size distribution.

3. We show that the recoverability of original set size distributions presents a sharp threshold with respect to the fraction of elements sampled from the sets. If this fraction lies below the threshold, then the original set size distribution or its average are unrecoverable. For large-scale power law networks, we need to sample more than 50% of the edges to obtain accurate estimates.

4. We show that the MLE achieves the error lower bound when the fraction of sampled elements is above the threshold.

5. We also derive lower bounds on the estimation error of the average set size.

1.1.3 Crawling a network to find target nodes

1. We introduce selective harvesting, the problem of searching networks to find a large a number of nodes that exhibit a specific trait as possible. We approach this problem by learning a model from node attributes and structural properties, while subject to a sampling budget. We show that state-of-the-art methods perform poorly in these problems because fitting a model to observations collected by the same model erodes performance over time, a phenomenon we dub *tunnel vision effect*.

2. We show that *classifier diversity* – i.e., alternating between several classifiers when deciding the next node to be sampled – can severely improve performance. Classifier diversity helps improve accuracy in two complementary ways. It explores more diverse regions and thus avoids remaining in a region where target nodes have been depleted. It also achieves *training sample diversity*, where diverse classifiers create enough diversity of observations to ease the *tunnel vision effect*.

3. We propose the Directed Diversity Dynamic Thompson Sampling (D^3TS), a new algorithm for deciding which classifier from a set to use during each step of selective harvesting. D^3TS is based on Multi-Armed Bandits for non-stationary reward distri-

butions, but in contrast to methods proposed for these problems, we enforce diversity by preventing the algorithm from converging to the use of a single classifier.

4. We evaluate the proposed framework on several real-world networks, including two datasets derived from donations to projects on Kickstarter.com and to projects on DonorsChoose.com. We observe that D^{3}TS' matches or exceeds the performance of the best method on each dataset.

1.2 Outline

The rest of this dissertation is organized as follows. In Chapter 2, we present definitions and review the background material needed in the following technical chapters. In Chapter 3 we investigate the problem of characterizing directed and undirected networks through RWs. In Chapter 4, we study – from an information-theoretic perspective – the set size distribution estimation problem. Next, in Chapter 5, we consider the problem of crawling a network to find target nodes by learning models from node attributes and structural properties. Last, in Chapter 6 we present some final remarks and future research directions.

CHAPTER 2

DEFINITIONS AND BACKGROUND

In this chapter we define terms and notation used throughout this dissertation. In addition, we provide an overview of sampling techniques for networks, which is relevant to Chapters 2 and 3. We also present the definition of Fisher information (a measure to quantify the statistical information in a sample) and explain how it relates to lower bounds on estimation errors through the Cramér-Rao bound. This definition and relationship are relevant to Chapter 3. Last, we provide an overview of Multi-Armed Bandit problems and algorithms, which is relevant to Chapter 4.

2.1 Definitions

Let $\mathcal{G}_d = (\mathcal{V}, \mathcal{E}_d)$ be a labeled directed graph representing the network topology, where $\mathcal{V}$ is a set of vertices (also called nodes) and $\mathcal{E}_d$ is a set of ordered pairs of vertices (u, v) representing a connection from u to v (a.k.a. edges). We refer to an edge (u, v) as an *in-edge* with respect to v and an *out-edge* with respect to u. The *in-degree* and *out-degree* of a vertex u in $\mathcal{G}_d$ are the number of distinct edges respectively into and out of u. We assume that each vertex in $\mathcal{G}_d$ has at least one edge (either an in-edge or an out-edge).

In some networks, all connections are reciprocal. In this case, we can represent such a network as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{E}$ is a set of unordered pairs of vertices (u, v) representing a connection between u and v. The degree of $v \in \mathcal{V}$, denoted by $\deg(v)$, is the number of pairs $(a, b) \in \mathcal{E}$ such that $a = v$. We can also represent such a network as a symmetric directed graph $\mathcal{G}_d$, i.e., $(u, v) \in \mathcal{E}_d$ iff $(v, u) \in \mathcal{E}_d$. In this case, the degree of $v \in V$ is defined as its in-degree or, equivalently, as its out-degree.

Let $\mathcal{L}$ be a finite (possibly empty) set of vertex labels. Let $\mathcal{L}(u) \subseteq \mathcal{L}$ denote the set of labels associated with vertex $u \in \mathcal{V}$. Each vertex $u \in \mathcal{V}$ is associated with a set of labels. For instance, one label $\ell \in \mathcal{L}(u)$ could be a country where an individual u has lived before.

A directed graph is composed of one or more strongly connected components. A *strongly connected component* (SCC) is the subgraph induced by a maximal set of nodes $\mathcal{C}$, such that for every pair $u, v \in \mathcal{C}$, there is a directed path from u to v and from v to u. If there is a path from one SCC $\mathcal{C}$ to another SCC $\mathcal{C}'$, there can be no path from $\mathcal{C}'$ to $\mathcal{C}$, by definition.

An undirected graph is composed of one or more connected components. A *connected component* (CC) is the subgraph induced by a maximal set of nodes $\mathcal{C}$, such that for every pair $u, v \in \mathcal{C}$, there is a path from u to v. There is no path between two distinct CCs.

2.2 Statistical Sampling of graphs

Sampling is vital to the characterization of large volumes of data. The idea is that, when it is impossible (or too expensive) to compute a statistic θ over the entire population which constitutes the data, we must sample units of this population and use statistical inference to estimate θ.

Important network characteristics include distributions related to (i) topology, such as degree distribution, clustering coefficient distribution, distribution of pair-wise distances, and to (ii) node traits – i.e., the fraction of individuals that have each trait. A sample is a sequence of observations consisting of nodes or edges from the network. An *estimator* is a function that takes a sequence of N observations $s_1, \ldots, s_N$ (sampled data) as input and outputs an estimate $\hat{\theta}$ of an unknown population parameter θ (graph characteristic).

When the probability of collecting each observation can be computed, estimators such as the Hansen-Hurwitz estimator can be used to approximate θ. Suppose θ is some average value computed over the entire population (e.g., average degree, average probability of being from a given country, etc). For all $i = 1, \ldots, N$, let S_i denote a random vari-

able corresponding to the i-th observation and π_i, $i = 1, \ldots, N$, denote the probability of observing (or sampling) S_i when collecting the i-th observation. The Hansen-Hurwitz estimator is given by

$$T(S_1, \ldots, S_N) = \frac{1}{N} \sum_{i=1}^{N} \frac{S_i}{\pi_i}. \tag{2.1}$$

Note that the estimator $T(S_1, \ldots, S_N)$ in (2.1) is a random variable that depends on the joint distribution of S_i, $i = 1, \ldots, N$. The most commonly used metric to assess the quality of an estimator is the *mean squared error* (MSE), given by

$$\text{MSE}(T(S_1, \ldots, S_N)) = E[(T(S_1, \ldots, S_N) - \theta)^2], \tag{2.2}$$

where the expectation is taken with respect to the joint distribution of $(S_1, \ldots, S_N)$. In some cases, the MSE can be derived analytically. In others, the expectation is approximated based on a large number of simulations using different seeds for the pseudo-random number generator.

Another useful metric to assess estimators is the *normalized root mean squared error* (NRMSE), defined as

$$\text{NRMSE}(T(S_1, \ldots, S_N)) = \frac{\sqrt{E[(T(S_1, \ldots, S_N) - \theta)^2]}}{\theta}. \tag{2.3}$$

The NRMSE is especially useful when comparing estimators of statistics that differ by one of more orders of magnitude. We consider NRMSE values as large as 1 to be acceptable regardless of the value of $\hat{\theta}$.

Sampling probabilities $\pi_1, \ldots, \pi_N$ depend on the sampling strategy in use. The sampling strategy depends, in turn, on the API available to query nodes and edges. In what follows, we discuss several strategies for sampling networks, indicating when they can be applied and how to compute sampling probabilities. For simplicity, we focus on undirected networks.

2.2.1 Uniform Vertex Sampling

In some networks, it is possible to perform uniform vertex sampling by generating node identifiers uniformly at random and checking whether each of them corresponds to a valid node or not. Unfortunately, in many cases (e.g., social networks) the id space is too sparse for this to be feasible. Uniform vertex sampling is still possible on networks that have an API to retrieve nodes uniformly (e.g., Wikipedia).

Uniform vertex sampling is typically used when we desire to estimate characteristics of the node population – e.g., degree distribution, fraction of nodes that have a certain node label. In that case, $\pi_i = 1/|\mathcal{V}|$ for all $i = 1, \ldots, N$. However, if we sample a node and then choose one of its edges uniformly as an observation, the probability of selecting an edge that connects nodes u and v is $\pi_i = (\frac{1}{\deg(u)} + \frac{1}{\deg(v)})/|\mathcal{E}|$.

2.2.2 Random Walk Sampling

On web graphs, we can visit a page to retrieve outgoing links to other pages. Similarly, on some social networks, we can visit a user profile to retrieve links to her acquaintances on the social network. In these cases, it is possible to perform a random walk (RW), a process in which a "particle" moves from node to node in a graph by traversing edges. At each step, the walker moves to a neighboring node chosen according to some probability distribution. In particular, if each neighboring node is chosen with equal probability, this process is referred as the uniform RW, or simply RW.

On an undirected graph, a RW defines a discrete time Markov chain that has steady state when the graph consists of a single connected component and is non-bipartite. In steady state, a RW samples edges according to a marginal distribution that is uniform on $\mathcal{E}$. Although the RW does not start in steady state and the observations it collects are not independent, most estimators assume so (or equivalently, that RW observations come from Uniform Edge Sampling). Therefore, the probability of traversing each edge is approxi-

mated by $\pi_i = 1/|\mathcal{E}|$. Moreover, the probability of observing (visiting) node $v \in \mathcal{V}$ is approximated by $\pi_i = 1/\deg(v)$.

2.2.3 Uniform Edge Sampling

Uniform edge sampling can be performed by sampling two nodes uniformly at random and checking whether an edge between them exists. In practice, it can only be used on very small networks. For each observation, the probability of sampling any given edge is $\pi_i = 1/|\mathcal{E}|$. When we choose one of the edge's adjacent nodes as an observation, the probability of sampling node v is $\pi_i = 1/\deg(v)$.

2.2.4 Independent Edge Sampling

Independent Edge Sampling can be thought of as a thinning process where we observe each edge from the graph with a fixed probability p. The total number of observed edges is binomially distributed with parameters $|\mathcal{E}|$ and p. This sampling process is useful to model a network where edges are hidden, but events taking place over some edges (e.g., message exchanges) make them observable. When the events over an edge are generated according to a homogeneous Poisson process with rate λ, the probability of observing each edge given an observation period Δt is given by $\pi_i \equiv p = \lambda \Delta t$.

2.2.5 Other sampling strategies

Other sampling techniques have been designed for collecting a subgraph that has similar characteristics (e.g., clustering coefficient) as the original graph [39, 52]. Typically, the probability of observing a given node or edge is unknown. These techniques are evaluated via simulation and often do not provide theoretical guarantees on the distance between measures of characteristics on the collected subgraph and on the original graph. For instance, in computer networks, the software tool `traceroute` can be run on a computer to sample all routers on a path between that machine and a given IP address.

2.3 Fisher information and Cramér-Rao Bound

Fisher information is one way to measure the statistical information contained in an observable random variable X about an unknown parameter θ that models the distribution of X. Let $f(X;\theta)$ be the likelihood function for θ, i.e., the probability density of the random variable X conditional on the value of θ. The Fisher information $J(\theta)$ is defined as the variance of the score function. The score function is the gradient of the natural logarithm of $f(X;\theta)$ w.r.t. parameter θ. Under weak regularity conditions [89, Chapter 2], the expected value of the score function is zero. Hence, the Fisher information is the second moment of the score, expressed as

$$J(\theta) = E\left[\left(\frac{\partial}{\partial\theta}\log f(X;\theta)\right)^2 \middle| \theta\right], \tag{2.4}$$

where the expectation is taken with respect to the distribution of X given θ, i.e., $f(X;\theta)$. The score function indicates how sensitively $f(X;\theta)$ depends on its parameter θ. When $\theta = (\theta_1, \ldots, \theta_W)$ is a vector, the Fisher information $J(\theta)$ is a matrix where its elements are given by

$$(J(\theta))_{i,j} = E\left[\left(\frac{\partial}{\partial\theta_i}\log f(X;\theta)\right)\left(\frac{\partial}{\partial\theta_j}\log f(X;\theta)\right) \middle| \theta\right]. \tag{2.5}$$

The Cramér-Rao Bound relates the Fisher information to the estimation error of any unbiased estimator $T(X) = (T_1(X), \ldots, T_W(X))$:

$$\text{cov}_\theta(T(X)) \succeq (J(\theta))^{-1}, \tag{2.6}$$

where $\text{cov}_\theta(T(X)) \succeq (J(\theta))^{-1}$ indicate that $\text{cov}_\theta(T(X)) - (J(\theta))^{-1}$ is a positive semi-definite matrix. From Threorem 5 in [42, Appendix A1.1.4], this implies that

$$(\text{cov}_\theta(T(X))_{ii} = \text{var}(T_i(X)) \geq ((J(\theta))^{-1})_{ii}. \tag{2.7}$$

2.4 Multi-Armed Bandits problems and algorithms

In Multi-Armed Bandit (MAB) problems, a forecaster is given a number of arms (or actions) K and a number of rounds T. For each round t, nature generates a payoff vector $\mathbf{r}_t = (r_{1,t}, \ldots, r_{K,t}) \in [0,1]^K$ unobservable to the forecaster.[1] The forecaster chooses an arm $I_t \in 1, \ldots, K$ and receives payoff $r_{I_t,t}$, with the other payoffs hidden. The goal is to maximize the cumulative payoff obtained.

MAB problems can be classified according to how the payoff vector is generated. In *stochastic bandit problems*, each entry $r_{i,t}$ in the payoff vector is sampled independently, from an unknown distribution ν_i, regardless of t. In *adversarial bandit problems*, the payoff vector $\mathbf{r}_t$ is chosen by an adversary which, at time t, knows the past, but not I_t. A popular algorithm for adversarial bandits is Exp3 (exponential-weight algorithm for exploration and exploitation) [7]. In essence, Exp3 selects an arm I_t probabilistically according to a weight vector, receives the reward associated to pulling I_t at time t and updates the weight of I_t proportionally to the observed reward scaled by the probability of selecting I_t. Exp3.P is a variant of Exp3 proposed by the same authors to offer better guarantees.

Stochastic and adversarial bandits do not cover the entire problem space, as the payoff vector distribution may vary over time in a less arbitrary way than in adversarial bandits. In *stochastic bandit problems with non-stationary distributions* or *dynamic bandit problems*, the mean payoff vector can evolve according to random shocks or change at pre-determined points in time.

MAB problems may also include context, which provides the forecaster with side-information about the optimal action at a given step. In *contextual bandits*, a context $\mathbf{x}_{a,t}$ is drawn (from some unknown probability distribution) for each action $a \in \mathcal{A}_t$ available in step t. The context may be provided explicitly or through recommendations of a set of experts. The recommendation is given as a probability distribution over the set of possible

[1] In general, rewards can be normalized to be in $[0,1]$.

actions. Auer et al. [7] proposes Exp4 (exponential-weight algorithm for exploration and exploitation using expert advice) to address this setting. Recommendations are combined proportionally to weights assigned to each expert. The observed reward is rescaled by the probability that the action is taken and experts weights are updated proportionally to the probability that each model select that node. Exp4.P is a variant of Exp4 proposed in [13] for which the same regret bounds hold with high probability. In [55], the authors propose LinUCB, a method that learns how to score actions assuming a linear relationship between the contexts and the observed payoffs. In [49], the authors propose the Epoch-Greedy algorithm, which has worse regret bounds than Exp4. However, when the number of hypotheses is infinite (but with finite VC-dimension) it has regret bounds, while Exp4 does not.

In *Restless Bandits* problems [92], each arm has an internal state that evolves according to an independent Markov chain. Even arms that were not pulled at time t make state transitions and output payoffs. The transition probability matrix and payoff may depend on whether the arm was pulled or not. The arm state is either observed or estimated after being pulled. Existing algorithms for Restless Bandits assume that the Markov chain that describes an arm is irreducible and composed by a few states.

CHAPTER 3

CHARACTERIZING DIRECTED AND UNDIRECTED NETWORKS VIA MULTIDIMENSIONAL WALKS WITH JUMPS

3.1 Introduction

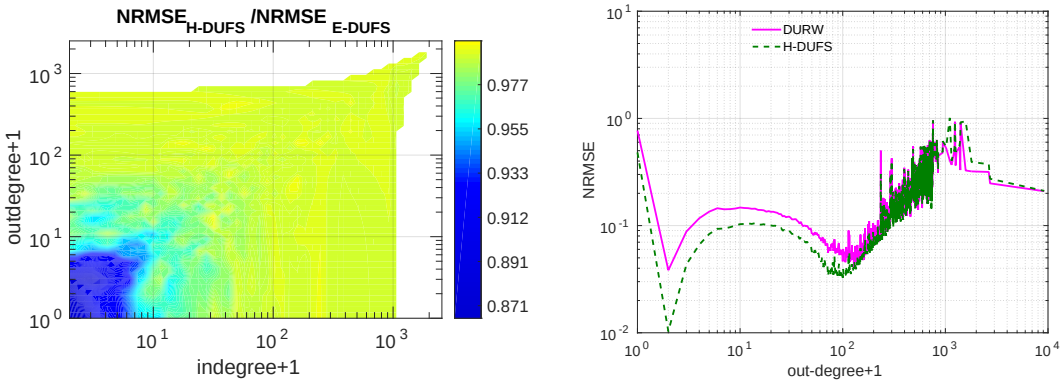
A number of studies [15, 18, 22, 29, 47, 52, 54, 62, 71, 73, 90] are dedicated to the characterization of complex networks. A complex network is a network with topological features that do not occur in simple networks such as lattices or random networks. Examples of such networks include the Internet, the Web, social, business, and biological networks. Characterizing a network consists of computing or estimating a set of statistics that describe the network. In this chapter we model a complex network as a directed or undirected graph with labeled vertices. A label can be, for instance, the degree of a vertex or, in a social network setting, someone's hometown. Label statistics (e.g., average, distribution) are often used to characterize a network.

Characterizing a network with respect to its labels requires querying vertices and/or edges; associated with each query is a resource cost (time, bandwidth, money). For example, information about web pages must be obtained by querying web servers while subject to a maximum query rate. Characterizing a large graph by querying the entire graph is often too costly. Even if the network is stored on disk it may constitute several terabytes of data. As a result, researchers have turned their attention to estimation of graph characteristics based on incomplete (sampled) data.

Simple strategies such as uniform vertex and uniform edge sampling possess desirable statistical properties: the former yields unbiased samples of the population and the bias introduced by the latter is easily removed. However, these strategies are often rendered

unfeasible because they require either a directory containing the list of all vertex (edge) ids, or an API that allows uniform sampling from the vertex (edge) space. Even when the space of possible vertex (edge) ids is known, its occupancy is usually so low that querying randomly generated ids is expensive. An alternate, cheaper, way to sample a network is via a random walk (RW). A RW samples a graph by moving a particle (walker) from a vertex to a neighboring vertex. It is applicable to any graph where we can query the edges connected to a given vertex. Furthermore, RWs share some of the desirable properties of uniform edge sampling (i.e., easy bias removal, accurate estimation of characteristics such as the tail of the degree distribution).

On one hand, a great deal of research has focused on designing sampling methods for *undirected graphs* using RWs [33, 71]. Ribeiro and Towsley proposed Frontier Sampling (FS), an n-dimensional random walk that uses n *coupled* random walkers [77]. This method yields more accurate estimates than the uniform RW and also outperforms the use of n independent walkers. In the presence of disconnected or loosely connected components, FS is even better suited than the uniform RW and independent RWs to sample the tail of the degree distribution of the graph. On the other hand, few works have focused on developing tools for characterizing *directed graphs* in the wild. A graph is said to be directed when edges are not necessarily reciprocated. Characterizing directed graphs through crawling becomes challenging when only outgoing edges from a node are visible (incoming edges are hidden): unless all vertices have a directed path to all other vertices, a walker will eventually be restricted to a (strongly connected) component of the graph. Furthermore, classic RWs incur biases that can only be removed by conditioning on the entire graph structure. Ribeiro et al. addressed these issues by proposing the Directed Unbiased Random Walk (DURW), a random walk sampling technique that performs degree-proportional jumps to obtain asymptotically unbiased estimates of the distribution of vertex labels on a directed graph [76].



(a) soc-Slashdot0902 (b) livejournal-links

Figure 3.1. Comparison between proposed method (DUFS) and previous state-of-the-art respectively for visible and for invisible incoming edges scenarios; (a) NRMSE ratios between DUFS $(w = 1, b = 10)$ and FS $(b = 10)$ of the estimated joint in- and out-degree distribution on the soc-Slashdot0902 dataset; (b) NRMSEs associated with DUFS and DURW of the estimated out-degree distribution on the livejournal-links dataset.

In this chapter, we propose Directed Unbiased Frontier Sampling (DUFS), which generalizes the FS and the DURW algorithms. Building on ideas in [76], we extend Frontier Sampling to allow the characterization of a network regardless of whether it is undirected, directed with observable incoming edges, or directed with unobservable incoming edges. DUFS matches or exceeds the performances of FS and DURW. This is illustrated in Figure 3.1. Simulation setup and datasets will be described in Section 3.5.1. [1]

3.1.1 Contributions

Our main contributions are as follows:

1. *Directed Unbiased Frontier Sampling (DUFS)*: we propose a new algorithm based on multiple coordinated random walks that extends Frontier Sampling (FS) to directed networks. DUFS generalizes FS [77] and DURW [76].

[1] In the figure caption, w and b are DUFS parameters corresponding to random jump weight and budget per walker.

2. *More accurate estimator for vertex label distribution*: the original estimator of vertex label distribution proposed for FS is based only on the observations collected during the walks. We introduce a new estimator that combines observations from initial walker locations with those made during the walks to produce better estimates. We show that these initial locations provide a valuable source of information regarding vertex labels associated with large probability masses.

3. *Practical recommendations*: we investigate the impact of the number of walkers and the probability of jumping to an uniformly chosen vertex (controlled via a parameter called random jump weight) on DUFS estimation errors, given a fixed sampling budget. By increasing the number of walkers the sequence of traversed edges approaches the uniform distribution faster, but this also increases the fraction of the budget spent to place the walkers in their initial locations. Moreover, increasing the random jump weight favors sampling vertex labels with large probability masses, which translates into more accurate estimates of the probability mass of these labels, but worse estimates for those in the tail. We study these trade-offs through simulation and propose guidelines for choosing DUFS parameters.

4. *Comprehensive evaluation*: we compare DUFS against other random walk-based methods w.r.t. estimation errors when applied on directed networks, both when incoming edges are directly observable and when they are not. When in-edges are observable, in addition to evaluating their performance in estimating the marginal in-degree and out-degree, distribution, some graph properties evaluated in previous works, we evaluate DUFS performance on estimating joint in- and out-degree distributions.

3.1.2 Outline

The estimation problem is stated in Section 3.2. In Section 3.3, we review FS and DURW algorithms. In Section 3.4, we propose the Directed Unbiased Frontier Sampling

(DUFS) algorithm (along with some estimators), which generalizes the previous methods. We investigate the impact of DUFS parameters on estimation accuracy in Section 3.5 and then provide practical guidelines on how to set them. A comparison with other random walk techniques is also provided. Section 3.6 discusses the NRMSE behavior on power law networks, DUFS' stopping criterion and the performance of DUFS in the absence of an API for obtaining uniform vertex samples. We discuss some related work and present our conclusions in Sections 3.7 and 3.8, respectively.

3.2 Problem Statement

Let $\mathcal{G}_d = (\mathcal{V}, \mathcal{E}_d)$ be a directed graph representing the network topology.[2] Let θ_ℓ, for $\ell \in \mathcal{L}_v$ be the fraction of nodes in V that is associated with label ℓ. The problem of estimating the node label distribution consists of estimating $\boldsymbol{\theta} = \{\theta_\ell\}_{\ell \in \mathcal{L}_v}$ from a set of observations. In this chapter, a set of observations consists of a set of uniform vertex samples and a set of vertices visited by RWs. There is a cost associated with a RW step, which is set to be 1. When sampling is done through uniform vertex sampling, the cost is taken to be $c \geq 1$ (typically larger). The cost c represents the difficulty of obtaining a valid node by querying randomly generated node ids. For instance, if only 5% of the id space is populated, $c = 20$. Sampling costs are taken off a budget denoted by B.

3.2.1 Input scenarios

When performing a RW, we assume that a walker retrieves the out-edges of the node where it resides by performing a query and that vertices are distinguishable. We define two scenarios depending on whether the walker can also retrieve in-edges. In the *first scenario* (both out- and in-edges can be retrieved) it is possible to move the walker over any edge regardless of the edge direction (if the edge is $(u, v) \in \mathcal{E}_d$ a walker can move from u to

[2]Undirected networks can also be represented this way by replacing each undirected edge with two directed edges.

v and from v to u). In this case, the walker can be seen as moving over $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, an undirected version of $\mathcal{G}_d$, i.e., $\mathcal{E} = \{(u, v) : (u, v) \in \mathcal{E}_d \vee (v, u) \in \mathcal{E}_d\}$. Define $\deg(v) = |\{(u, v) : (u, v) \in \mathcal{E}\}|$. Let $\text{vol}(S) = \sum_{\forall v \in S} \deg(v), \forall S \subseteq \mathcal{V}$, denote the volume of the set of vertices in $S \subseteq \mathcal{V}$.

In the *second scenario* (only out-edges are directly observable), we can build on-the-fly an undirected graph $\mathcal{G}_u$ based on the out-edges that have been sampled. Details on how to construct $\mathcal{G}_u$ will be presented in the following sections. For now, note that $\mathcal{G}_u$ is not an undirected version of $\mathcal{G}_d$ as some of the in-edges of a node may not have been observed. By moving the walker over $\mathcal{G}_u$ – possibly traversing edges in $\mathcal{G}_d$ in the opposite direction – we can compute its stationary behavior and thus, remove the bias by accounting for the probability that each observation appears in the sample.

3.3 Background

In what follows, we review a representative RW-based method proposed for each of the two scenarios proposed in Section 3.2. First, we describe Frontier Sampling [77], an algorithm that relies on n coupled random walks. This technique can be applied to undirected graphs and to directed graphs provided that both incoming and outgoing edges at a node are observable. Then, we describe the Directed Unbiased Random Walk [76], an algorithm that adapts a single random walk to directed graphs when incoming edges are not directly observable. The goal of these methods is to obtain samples from a graph, which are then used for inferring graph characteristics via an estimator.

Although not required, in the first scenario, it is useful to keep track of the observed graph during the sampling process. Storing information about visited nodes in memory saves resources that would be consumed to query those nodes in subsequent visits – i.e., revisiting a node has no cost. In the second scenario, we need to store a variant of the observed graph. This variant will be described in Section 3.3.2. As the budget B grows larger, the space needed to store the observed graph tends to $O(B\bar{d})$, where $\bar{d}$ is the average

Algorithm 1 Frontier Sampling (uniform vertex sampling cost c, budget per walker b)

1: $n \leftarrow B/(c+b)$
2: $i \leftarrow 0$ $\{i$ is step counter$\}$
3: Initialize $L = (v_1, \ldots, v_n)$ with n randomly chosen vertices (uniformly)
4: **do**
5: Select $u \in L$ with probability $\deg(u) / \sum_{\forall v \in L} \deg(v)$
6: Select an edge (u, v), uniformly at random
7: Replace u by v in L and add (u, v) to sequence of sampled edges
8: $i \leftarrow i + 1$
9: **while** $i \geq B - nc$

out-degree of the underlying graph. For small values of B, the space complexity depends on how the average degree of the observed graph grows with the number of samples and is left for future investigation.

3.3.1 Frontier Sampling: a multidimensional random walk for undirected networks

In essence, *Frontier Sampling* (FS) is a random walk-based algorithm for sampling and estimating characteristics of an undirected graph. FS performs n *coordinated* random walks on the graph. One of the advantages of using multiple walkers is that they can cover multiple connected components (when they exist), while a single walker is restricted to one component in the absence of a random jump or restart mechanism. However, when random walks are independent (not coordinated) the number of samples obtained from a component is proportional to the number of walkers in that component. Therefore, the probability of sampling an edge in steady state will differ for different components, unless the number of walkers in each component is set to be proportional to its volume. Unfortunately, initializing the walkers in such a way requires knowing the component volumes in advance, which cannot be done in practice. By coordinating multiple random walkers, FS is able to sample edges uniformly at random in steady state regardless of how the walkers are initially placed.

Algorithm 1 describes FS. There are two parameters, the initial cost of placing a walker $c \geq 1$ and the average number of new nodes sampled by a walker b. The initial walker

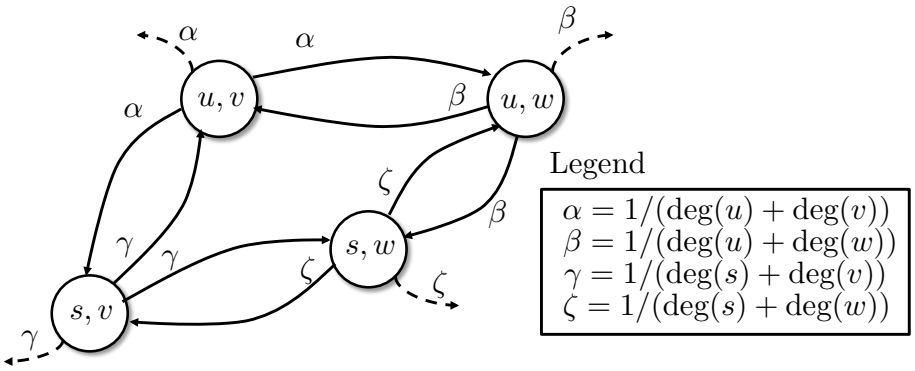


Figure 3.2. Illustration of the Markov chain associated to FS with dimension $n = 2$ (adapted from [77]).

locations are chosen uniformly at random over the vertex set. Note that the number of walkers is taken to be $n = \lfloor B/(c+b) \rfloor$, that the cost of taking a random walk step is one (except for previously sampled nodes) and that the cost of initially placing a walker, c, can be greater than one because uniform vertex sampling is often expensive. FS maintains a list L of n vertices representing the locations of the n walkers. At each step, a walker is chosen from L in proportion to the degree of the node where it is currently located. The walker then moves from u to an adjacent vertex v.

The Frontier sampling process is equivalent to the sampling process of a single random walk over the n-th Cartesian power of $\mathcal{G}$, $\mathcal{G}^n = (\mathcal{V}^n, \mathcal{E}_n)$, where

$$\mathcal{V}^n = \{(v_1, \ldots, v_n) \,|\, v_1 \in \mathcal{V} \wedge \cdots \wedge v_n \in \mathcal{V}\}$$

is the n-th Cartesian power of $\mathcal{V}$. For all $\mathbf{v}, \mathbf{u} \in \mathcal{V}^n$, $(\mathbf{v}, \mathbf{u}) \in \mathcal{E}_n$ if there exists an index $i \in \{1, \ldots, n\}$ such that $(v_i, u_i) \in \mathcal{E}$ and $u_j = v_j$ for $j \in \{1, \ldots, n\}/\{i\}$ [77, Lemma 5.1]. For this reason, Frontier Sampling can be thought of as an n-dimensional random walk (see Fig. 3.2).

Let $L_t = (v_1, \ldots, v_n)$ denote the state of FS before the t-th step, $t = 1, \ldots$. Theorem 3.3.1 establishes key statistical properties of Frontier Sampling. A more complete version of this theorem is presented and proved in [77, Theorem 5.2].

Theorem 3.3.1. *Recall that $\mathcal{G}$ is an undirected graph. If $\mathcal{G}$ is connected and non-bipartite, then the stationary behavior FS exhibits the following properties:*

(I) sampled edges form a stationary sequence and their marginal distribution is uniform on $\mathcal{E}$,

(II) $L_\infty = (v_1, \ldots, v_n)$ has the unique distribution

$$\pi_{\mathbf{v}} = \frac{\sum_{i=1}^{n} \deg(v_i)}{n|\mathcal{V}|^{n-1} vol(\mathcal{V})}, \quad \textit{for } \mathbf{v} \in \mathcal{V}^n.$$

Using FS samples to estimate vertex label distributions is simple when the input corresponds to the first scenario described in Section 3.2. The probability of sampling a given node is proportional to its undirected degree in $\mathcal{G}$. Hence, each sample must be weighted inversely proportional to the respective node's undirected degree. Storing the undirected version of the observed graph along with labels associated with sampled nodes allows the sampler to avoid having to pay the cost of revisiting a node.

Conversely, when incoming edges are not observed, there is a less straightforward way to adapt Frontier Sampling, which we propose in Section 3.4.

3.3.2 Directed Unbiased Random Walk: a random walk adapted for directed networks with unobservable in-edges

The presence of hidden incoming edges but observable outgoing edges makes characterizing large directed graphs through crawling challenging. Edge (u, v) is a hidden incoming edge of node v if (u, v) can only be observed from node u. For instance, in Wikipedia we cannot observe the edge ("Columbia Records", "Thomas Edison") from Thomas Edison's wiki entry (but this edge is observable if we access the Columbia Records's wiki entry).

These hidden incoming edges make it impossible to remove the bias incurred by walking on the observed graph, unless we crawl the entire graph. Moreover, there may not

even be a directed path from a given node to all other nodes. Graphs with hidden outgoing edges but observable incoming edges exhibit essentially the same problem. In [76], we proposed the Directed Unbiased Random Walk (DURW) algorithm, which obtains asymptotically unbiased estimates of vertex label densities on a directed graph with unobservable incoming edges. Our random walk algorithm resorts to two principles to achieve unbiased samples and reduce variance:

- *Backward edge traversals*: in real-time we construct an undirected graph $\mathcal{G}_u$ using the nodes that are sampled by the walker on the directed graph $\mathcal{G}_d$. The role of the undirected graph is to guarantee that, at the end of the sampling process, we can approximate the probability of sampling a node, even though in-edges are not observed. The random walk proceeds in such a way that its trajectory on $\mathcal{G}_d$ is consistent with that of a random walk on $\mathcal{G}_u$. The walker is allowed to traverse some of the edges in $\mathcal{G}_d$ in a reverse direction. However, we prevent some of the observed edges to be traversed in the reverse direction by not including them in $\mathcal{G}_u$. More precisely, once a node v is visited at the i-th step, no in-edges to v observed at step $j > i$ (by visiting nodes w such that $(w, v) \in \mathcal{E}_d$) are added to $\mathcal{G}_u$. This restriction leads to Lemma 3.1 below. The fact that the degree of visited nodes in G_u remains fixed is an important feature to reduce the random walk transient and thus, reduce estimation errors.

- *Degree-proportional jumps*: the walker makes a limited number of random jumps to guarantee that different parts of the directed graph are explored. In DURW, the probability of randomly jumping out of a node v, $\forall v \in \mathcal{V}$, is $w/(w + \deg(v))$, $w > 0$. This modification is based on the following observation: let $\mathcal{G}_u$ be a weighted undirected graph formed by adding a *virtual* node σ such that σ is connected to all nodes in $\mathcal{V}$ with edges having weight w. All remaining edges have unit weight. In a weighted graph a walker transverses a given edge with probability proportional to the weight of this edge. The steady state probability of visiting a node v on $\mathcal{G}_u$ is

$(w + \deg(v))/(\text{vol}(\mathcal{V}) + w|\mathcal{V}|)$. Similar to the cost of placing a FS walker through uniform vertex sampling, each random jump incurs cost c.

Lemma 3.1. *The degree of a node v in $\mathcal{G}_u$ does not change after v is visited by the first walker.*

The proof follows by definition from the backward edge traversal principle.

3.3.2.1 The DURW algorithm

DURW is a random walk over a *weighted undirected connected graph* $\mathcal{G}_u = (\mathcal{V}, \mathcal{E}_u)$, which is built on-the-fly. We build an undirected graph using the underlying directed graph $\mathcal{G}_d = (\mathcal{V}, \mathcal{E}_d)$ and the ability to perform random jumps. Let $\mathcal{G}^{(i)} = (\mathcal{V}, \mathcal{E}^{(i)})$ denote the undirected graph constructed by DURW at step i, where $\mathcal{V}$ is the node set and $\mathcal{E}^{(i)}$ is the edge set. Denote by $\mathcal{G}_u \equiv \lim_{i\to\infty} \mathcal{G}^{(i)}$. In what follows we describe the construction of $\mathcal{G}^{(i)}$.

Let $\mathcal{N}(v)$ denote the set of out-edges of a node v in $\mathcal{G}_d$. To simplify our exposition, we include a virtual node σ in the constructed graph, which represents a random jump. Let $\mathcal{S}^{(i)} = \{s_1, \ldots, s_i\}$ be the set of nodes from $\mathcal{V} \cup \{\sigma\}$ sampled by the random walk up to step i, where s_j denotes the node on which the walker resides at step j. The walker starts at node $s_1 \in \mathcal{V}$. We initialize $\mathcal{G}^{(1)} = (\mathcal{V}, \mathcal{E}^{(1)})$, where $\mathcal{E}^{(1)} = \mathcal{N}(s_1) \cup \{(u, \sigma) : \forall u \in \mathcal{V}\}$, where $\{(u, \sigma) : \forall u \in \mathcal{V}\}$ is the set of all undirected virtual edges to node σ. Let

$$W(u, v) = \begin{cases} w, & \text{if } u = \sigma \text{ or } v = \sigma \\ 1, & \text{otherwise} \end{cases}$$

denote the weight of edge (u, v), $\forall (u, v) \in \mathcal{E}^{(i)}$. The next node, s_{i+1}, is selected from $\mathcal{E}^{(i)}$ with probability $W(s_i, s_{i+1}) / \sum_{\forall (s_i, v) \in \mathcal{E}^{(i)}} W(s_i, v)$. In case the selected node is σ, the walker immediately moves to a node v uniformly chosen from $\mathcal{V}$, which then becomes s_{i+1}.

These two node selections (σ and v) altogether incur cost c. Upon selecting s_{i+1} we update $\mathcal{G}^{(i+1)} = (\mathcal{V}, \mathcal{E}^{(i+1)})$, where

$$\mathcal{E}^{(i+1)} = \mathcal{E}^{(i)} \cup \mathcal{N}'(s_{i+1}), \tag{3.1}$$

and

$$\mathcal{N}'(s_{i+1}) = \{(s_{i+1}, v) : \forall (s_{i+1}, v) \in \mathcal{N}(s_{i+1}) \text{ s.t. } v \notin \mathcal{S}^{(i)}\}$$

is the set of all edges (u, v) in $\mathcal{N}(s_{i+1})$ where node $v \notin \mathcal{S}^{(i)}$. Note that $\mathcal{N}'(s_{i+1}) \subseteq \mathcal{N}(s_{i+1})$. By using $\mathcal{N}'(s_{i+1})$ instead of $\mathcal{N}(s_{i+1})$ in equation (3.1) we guarantee that no node in $\mathcal{S}^{(i)}$ changes its degree, i.e., $\forall v \in \mathcal{S}^{(i)}$ the degree of v in $\mathcal{G}^{(i)}$ is also the degree of v in $\mathcal{G}_u$. Thus, we comply with the requirement that once a node v, $\forall v \in \mathcal{V}$, is visited by the RW no edge can be added to $\mathcal{G}_u$ with v as an endpoint.

In the actual implementation, it is only necessary to keep track of nodes in $\mathcal{S}^{(i)} \cup \bigcup_{v \in \mathcal{S}^{(i)} \setminus \{\sigma\}} \mathcal{N}(v)$ and the edges in $\mathcal{E}_d$ leaving each node $v \in \mathcal{S}^{(i)} \setminus \{\sigma\}$. In fact, while the virtual node σ is connected to all nodes in $\mathcal{V}$, the sampler does not have access to the identities of nodes other than the ones that were already observed. In order to estimate vertex label distributions from DURW observations, we weight samples in proportion to the reciprocal of the probability that the corresponding vertices are visited by a random walk in $\mathcal{G}_u$, in steady state. Storing the labels associated with nodes in $\mathcal{S}^{(i)} \setminus \{\sigma\}$ saves the cost of querying repeated nodes.

3.4 Generalizing FS: a new method applicable regardless of in-edges visibility

This section is divided into two parts. In Section 3.4.1 we propose the Directed Unbiased Frontier Sampling (DUFS) method, which generalizes FS to allow estimation on directed graphs with unobservable in-edges (second scenario described in Section 3.2). DUFS also generalizes DURW to provide the benefits obtained from using multiple cou-

pled walkers. DURW is a special case of DUFS where the number of walkers is one. Next, in Section 3.4.2, we describe two ways to estimate node label distributions from DUFS samples. The first ignores observations from the initial walker placement and uses only on observations collected during the random walks. The second estimator leverages the information contained in the initial walker locations in addition to subsequent walker steps.

3.4.1 Directed Unbiased Frontier Sampling

Like FS, Directed Unbiased Frontier Sampling (DUFS) samples a network through n coordinated walks. Similar to DURW, it constructs an undirected graph $\mathcal{G}_u = (\mathcal{V}, \mathcal{E}_u)$ in real-time that allows *backward edge traversals*. Denote by $\mathcal{G}^{(i)} = (\mathcal{V}, \mathcal{E}^{(i)})$ the undirected graph constructed by DUFS at step i. At each step, it selects a walker in proportion to the degree of the node where it currently resides. After a walker visits vertex $u \in \mathcal{V}$ for the first time, DUFS includes all out-edges from u in $\mathcal{G}^{(i)}$, except the ones that can cause walkers to have a view of the graph that is inconsistent with the view at a previous point in time. In other words, when node u is visited for the first time at step i, u is inserted in $\mathcal{G}^{(i)}$ along with all edges $(u, v) \in \mathcal{E}_d$ such that v has not been sampled. Thus, the degree of u is fixed in $\mathcal{G}^{(j)}$, for all $j \geq i$.

It may seem that there is no need to include *degree-proportional jumps* to visit different graph components when a large number of walkers are initially spread throughout the graph (e.g., on vertices chosen uniformly at random). However, including degree-proportional jumps in DUFS is still beneficial because it prevents walkers from being trapped when initially located on vertices whose out-degree is zero. More generally, it allows walkers to move from small volume to large volume components and, hence, obtain more samples among large degree nodes.

Algorithm 2 gives a high-level pseudo-code description of DUFS. At each step i, DUFS needs to keep track of $\mathcal{G}^{(i)}$ for $i = 1, \ldots$. In the extreme case where $n = B/c$, walkers are initialized but no budget is left to perform steps (i.e., $b = 0$). Thus, DUFS degenerates to

Algorithm 2 Directed Unbiased Frontier Sampling (budget per walker b, random jump weight w)

$n \leftarrow B/(c+b)$ ▷ n is the number of walkers
$i \leftarrow 0$ ▷ i is the current number of steps
Initialize $L = \{v_1, \ldots, v_n\}$ with n randomly chosen vertices (uniformly)
do
    Select $v \in L$ with probability $\frac{w+\deg(v)}{nw+\sum_{\forall v_j \in L} \deg(v_j)}$
    Sample $p \sim \text{Uniform}(0,1)$
    if $p < \frac{w}{w+\deg(v)}$ **then**
        Select a vertex $v \in \mathcal{V}$ uniformly at random
    else
        Select an outgoing edge of v, (v, v'), uniformly at random
        Replace v by v' in L and add (v, v') to sequence of sampled edges
        $i \leftarrow i + 1$
while $i \geq B - nc$

uniform vertex sampling. When the underlying graph is symmetric and the jump weight is $w = 0$, it becomes FS.

3.4.2 Vertex Label Distribution Estimation

In this section we describe two estimators of vertex label distributions from samples obtained by DUFS. The same estimators also apply to FS and DURW samples. For a description of estimators of edge label distribution and other graph characteristics, please refer to [77].

3.4.2.1 The edge-based estimator

Let s_i denote the i-th node visited by DUFS, $i = 1, \ldots$. Let θ_ℓ denote the fraction of nodes in $\mathcal{V}$ with label $\ell \in \mathcal{L}$. The steady state probability of sampling node v in $\mathcal{G}^{(t)}$ is given by

$$\pi(v) = \frac{w + \deg(v)}{|\mathcal{V}|w + \sum_{u \in \mathcal{V}} \deg(u)}, \quad \forall v \in \mathcal{V},$$

where $\deg(v)$ is the degree of v in $\mathcal{G}^{(t)}$. The vertex label distribution is estimated at t as

$$\hat{\theta}_\ell = \frac{1}{n} \sum_{i=1}^{t} \frac{\mathbb{1}\{\ell \in \mathcal{L}(v)\}}{\hat{\pi}(s_i)}, \qquad \ell \in \mathcal{L},\, t = 1, \ldots, \tag{3.2}$$

where $\mathbb{1}\{P\}$ takes value one if predicate P is true and zero otherwise, and $\hat{\pi}(s_i)$ is an estimate of $\pi(s_i)$: $\hat{\pi}(s_i) = (w + \deg(s_i))S$. Here

$$S = \frac{1}{t} \sum_{i=1}^{t} \frac{1}{w + \deg(s_i)} \,. \tag{3.3}$$

The following theorem states that $\hat{\pi}(s_i)$ is asymptotically unbiased.

Theorem 3.2. *$\hat{\pi}(s_i)$ is an asymptotically unbiased estimator of $\pi(s_i)$.*

Proof. To show that $\hat{\pi}(s_i)$ is asymptotically unbiased, we first note that the limit $\lim_{t\to\infty} \mathcal{E}^{(t)} = \mathcal{E}^{(\infty)}$ exists, since it visits all vertices w.h.p., after which no additional edges are included. We then invoke Theorem 4.1 of [77], yielding $\lim_{t\to\infty} S = |\mathcal{V}|/(|\mathcal{E}^{(\infty)}| + |\mathcal{V}|w)$ almost surely. Thus, $\lim_{t\to\infty} \hat{\pi}(s_i) = \pi(s_i)$ almost surely. Taking the expectation of (3.2) in the limit as $t \to \infty$ yields $\mathcal{E}[\lim_{t\to\infty} \hat{\theta}_\ell] = \theta_\ell$, which concludes our proof.

3.4.2.2 The hybrid estimator: leveraging information from initial walker locations

Note that the estimator presented in (3.2) does not make use of information associated with the initial set of nodes on which the walkers are placed. When the number of walkers is large this results in the loss of a considerable amount of statistical information. However, including these observations is challenging because subsequent observations from RW steps are not independent of the initial observations. Moreover, the normalizing constant for the random walk observations is no longer given by (3.3), since the degree distribution estimates also depend on the information contained in the random vertex samples.

In this section, we derive a new estimator that circumvents these problems by approximating the likelihood of random walk samples by that associated with uniform edge sampling. We call it the *hybrid estimator* because it combines observations from initial walker locations and the random walks. The hybrid estimator significantly improves the estimation accuracy for labels associated with large probability masses.

For simplicity, assume that each vertex has exactly one label. Let us index the vertex labels $\mathcal{L}$ from 1 to W, where $W = |\mathcal{L}|$. Denote the vertex label distribution by $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_W)$. Let n_i denote the number of walkers starting on label i nodes and $m_{i,j}$ the number of subsequent observations of nodes that have label i and undirected degree j. We approximate random walk samples in DUFS by uniform edge samples from $\mathcal{G}_u$. Denote by Z the maximum undirected degree in the underlying graph $\mathcal{G}$. Experience from previous studies shows us that this approximation works very well in practice. Hence, the likelihood function given the samples $\mathbf{n} = \{n_i : i = 1, \ldots, W\}$ and $\mathbf{m} = \{m_{i,j} : i = 1, \ldots, W \text{ and } j = 1, \ldots, Z\}$ is expressed as

$$L(\boldsymbol{\theta}|\mathbf{n}, \mathbf{m}) = \frac{\prod_i \theta_i^{n_i} \prod_k ((w+k)\theta_{i,k})^{m_{i,k}}}{\left(\sum_{s,t}(w+t)\theta_{s,t}\right)^M}. \tag{3.4}$$

The maximum likelihood estimator $\theta^\star$ is the value of θ that maximizes (3.4) subject to $0 \leq \theta_i \leq 1$ and $\sum_i \theta_i = 1$. This defines a constrained non-convex optimization problem. We transform this optimization problem into an unconstrained non-convex problem using the reparameterization $\theta_i = e^{\beta_i} / \sum_k e^{\beta_k}$ for $i = 1, \ldots, W$. As shown in Appendix A, the partial derivatives of the resulting objective function are given by

$$\frac{\partial \mathcal{L}(\boldsymbol{\beta}|\mathbf{n}, \mathbf{m})}{\partial \beta_i} = n_i + m_i - \frac{N e^{\beta_i}}{\sum_j e^{\beta_j}} - \frac{M e^{\beta_i} m_i/\mu_i}{\sum_s e^{\beta_s} m_s/\mu_s}, \qquad i = 1, \ldots, W, \tag{3.5}$$

where $m_i = \sum_k m_{i,k}$ and $\mu_i = \sum_k m_{i,k}/(w+k)$. Setting one of the variables to a constant (say, $\beta_W = 1$) for identifiability and then using the gradient descent procedure to change the remaining variables according to (3.5) is guaranteed to converge provided that we make small enough steps. However, there is no guarantee that it converges to the true global maximum.

An interesting interpretation of (3.5) is obtained by setting the derivatives to zero and substituting back $\theta_i = e^{\beta_i} / \sum_k e^{\beta_k}$:

$$\theta_i^\star = \frac{n_i + m_i}{N + M\frac{m_i/\mu_i}{\sum_s \theta_s^\star m_s/\mu_s}}, \qquad i = 1, \ldots, W. \tag{3.6}$$

According to (3.6), the estimated fraction of nodes with label i is the total number of times label i was observed (i.e., $n_i + m_i$) normalized by sum of (i) the number of uniform vertex samples and (ii) the number of uniform edge samples weighted by the probability of sampling label i from one uniform edge sample. In the limit as N and M go to infinity, we can show that $\boldsymbol{\theta}^\star = \boldsymbol{\theta}$ is a solution, but we cannot prove that it is unique or that $\boldsymbol{\theta}^\star$ converges to $\boldsymbol{\theta}$. Hence, we cannot prove that $\boldsymbol{\theta}^\star$ is asymptotically unbiased.

The system of non-linear equations determined by (3.6) cannot be solved directly, but can be estimated by Expectation Maximization (EM). In this case, the term $\sum_s \theta_s^\star m_s/\mu_s$ in the denominator is replaced by its expected value given θ_i's from the previous iteration. Based on the same idea, if we replace $\sum_s \theta_s^\star m_s/\mu_s$ with an edge sampled-based estimator $\hat{d}$, we obtain the following non-recursive variant of the hybrid estimator,

$$\hat{\theta}_i = \frac{n_i + m_i}{N + M\frac{m_i}{\mu_i \hat{d}}}, \qquad i = 1, \ldots, W, \tag{3.7}$$

where $\hat{d} = M/(\sum_i \mu_i)$. Theorem 3.4.1 below states the conditions under which $\hat{\theta}_i$ is asymptotically unbiased (see Appendix A for proof). In practice, we find no significant difference between $\theta_i^\star$ and $\hat{\theta}_i$, except when the number of walkers N is very large and the jump weight w is very small. For those cases, $\theta_i^\star$ tends to be slightly more accurate than $\hat{\theta}_i$ for small values of i, which in some applications may justify the additional computational cost of executing gradient descent or EM.

Theorem 3.4.1. *Let $N = \alpha B$ and $M = (1-\alpha)B$, for some $0 < \alpha < 1$. In the limit as $B \to \infty$, $\hat{\theta}_i$ is an unbiased estimate of θ_i.*

In the special case where the label is the undirected degree itself, we have $\mu_i = m_i/(w+i)$. Hence, eq. (3.7) reduces to

$$\bar{\theta}_i = \frac{n_i + m_i}{N + M(w+i)/\hat{d}}, \tag{3.8}$$

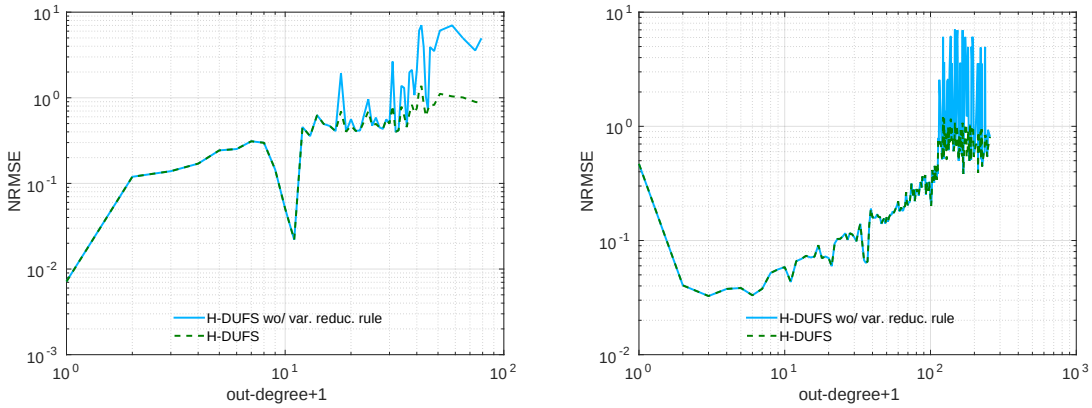
where $\hat{d}$ is the estimated average degree. When the average degree is known, we can show that $\bar{\theta}_i$ is the minimum variance unbiased estimator (MVUE) of θ_i (see Appendix A for proof).

When $n_i > 0$ but $m_i = 0$, the estimator in eq. (3.7) reduces to $\hat{\theta}_i = n_i/N$, which is essentially the MLE for uniform vertex sampling. It is well known that this estimator is not nearly as accurate as a RW-based estimator for large out-degree values with small probability mass. In some sense, the estimator $\hat{\theta}_i = n_i/N$ does not account for the fact that the number of RW samples is zero. As a result, mass estimates for large out-degrees tend to have very large variance when no RW samples are observed. Fortunately, we find that the following heuristic rule can drastically reduce the estimator variance in these cases.

3.4.2.3 Variance reduction rule

If no uniform edge samples are observed for out-degree i, we set the estimate $\hat{\theta}_i = 0$. This implies that we ignore any uniform vertex samples seen of nodes that have out-degree i. While this obviously results in a biased estimate, as the budget per walker b goes to infinity, the probability that this rule is invoked goes to zero. Hence, this rule produces an asymptotically unbiased estimate. This rule can be interpreted as a combination of vertex-based and edge-based estimates in proportion to the reciprocals of their estimated variances. That is, when no uniform edge samples are observed for a given out-degree, the corresponding estimated variance is zero and hence, uniform vertex samples should be ignored. We note that the converse rule (i.e., setting $\hat{\theta}_i = 0$ if no uniform vertex samples were observed) would not perform well, as the probability of sampling large out-degrees with uniform vertex sampling is very small.

We simulate DUFS on several datasets and compare the results obtained with the hybrid estimator when the rule is used and when it is not. Simulation details and datasets will be described in Section 3.5.1. Figures 3.3(a-b) show typical results of the impact of the



(a) p2p-Gnutella31 (b) web-Stanford

Figure 3.3. (visible in-edges) Effect of variance reduction rule on NRMSE, when $B = 0.1|\mathcal{V}|$ and $c = 1$. Using information contained in uniform vertex samples can increase variance for large out-degree estimates. However, the proposed rule effectively controls for that effect without decreasing head estimates accuracy.

rule when estimating out-degree distributions using DUFS in conjunction with the hybrid estimator on two network datasets (averaged over 1000 runs). The results show that the rule consistently reduces estimation error in the distribution tail without affecting estimation quality for small values of i.

3.4.2.4 In-degree distribution: impossibility result

The fact that long random walks are often approximated by uniform edge sampling brings up the question of whether they can be used to estimate in-degree distributions when the in-degree is not observed directly. Under uniform edge sampling, the number of observed edges pointing to a node is binomially distributed and a maximum likelihood estimator can be derived for estimating the in-degree distribution. This problem is related to the set size distribution estimation problem, where elements are randomly sampled from a collection of non-overlapping sets and the goal is to recover the original set size distribution from samples. In addition to in-degree distribution in large graphs, this problem is related to the uncovering of TCP/IP flow size distributions on the Internet.

We investigate this problem in Chapter 4, where we derive error bounds for the set size distribution estimation problem from an information-theoretic perspective. The recoverability of original set size distributions present a sharp threshold with respect to the fraction of elements sampled from the sets. If this fraction lies below the threshold, typically half of the elements in power-law and heavier-than-exponential-tailed distributions, then the original set size distribution is unrecoverable.

3.5 Results

This section is divided into three parts. First, we investigate the impact of DUFS parameters on estimation accuracy. We then compare DUFS against other random walk-based methods when both outgoing and incoming edges are visible. Finally, we perform a similar comparison when only out-edges are visible. We refer to the edge-based estimator defined in (3.2) and the hybrid estimator defined in (3.6) as E-DUFS and H-DUFS respectively.

In our evaluation, we simulate each method on 15 datasets taken from the Stanford SNAP repository [53] that describe topologies of different directed networks. These datasets correspond to a variety of social networks, communication networks, web graphs, one Internet peer-to-peer networks and one product co-purchasing networks. We find it informative to extract the largest strongly connected component of each network and to apply our methods to the resulting datasets, which we refer to as LCC datasets. Figure 3.4 shows the out-degree probability mass function (p.m.f.) for each network, along with the out-degree p.m.f. for the corresponding LCC dataset. We opt to show the p.m.f. instead of the complementary cumulative distribution function (CCDF) because the estimation task in this chapter is defined in terms of the p.m.f.'s. Defining the estimation task in terms of the CCDF would give H-DUFS an unfair advantage, as we will see in Section 3.5.2.

Simulations consist of sampling the graph until a budget $B = 0.1|\mathcal{V}|$ (i.e., 10% of the number of vertices) is depleted. Note that budget is decremented when walkers are initially placed and each time one of them moves to a vertex and when they perform random jumps.

We construct an undirected graph in the background throughout each simulation. As a result, we assume that the cost to revisit a vertex is zero, even if this visit occurs due to a random jump.[3]

When both outgoing and incoming edges are observable, random walks disregard edge direction, and move as if the network was undirected. In this scenario, we focus either on the estimation of the marginal out- and in-degree distributions or the joint distribution. The methods we investigate here can be used to estimate other node label distributions. For instance, if the underlying network is undirected, we can estimate the (undirected) degree distribution or even non-topological properties, such as the distribution of user nationalities in a social network. In the light of the impossibility results described in the end of Section 3.4.2, we focus on out-degree distribution estimation when incoming edges are not directly observable.

In the case of marginal in-degree (out-degree) distribution, we refer to in-degrees (out-degrees) smaller than the average as the *head* of the distribution. We refer to the top 1% largest in- (out-degree) values as the *tail* of the distribution.

3.5.1 Impact of DUFS parameters and practical guidelines

To provide some intuition on how the random jump weight w and the budget per walker b affect the accuracy of DUFS estimates, assume for now that we replace samples collected via random walks by uniform edge samples from the weighted undirected graph $\mathcal{G}_u$. In this hypothetical scenario, the budget B is used to collect $n \geq 1$ uniform vertex samples and $B - nc$ uniform edge samples. Clearly, when the edge-based estimator defined in (3.2) is used, the most accurate vertex label distribution estimates are obtained by setting $n = 1$, or equivalently, $b = B - c$. Therefore, we focus on the case where the hybrid-estimator defined in (3.6) is used. In particular, consider estimation of the out-degree distribution.

[3]Note that the alternative, i.e. always taking c units off the budget per random jump, is unlikely to impact results significantly when $B = 0.1|\mathcal{V}|$, since the vast majority of random jumps will find a non-visited node.

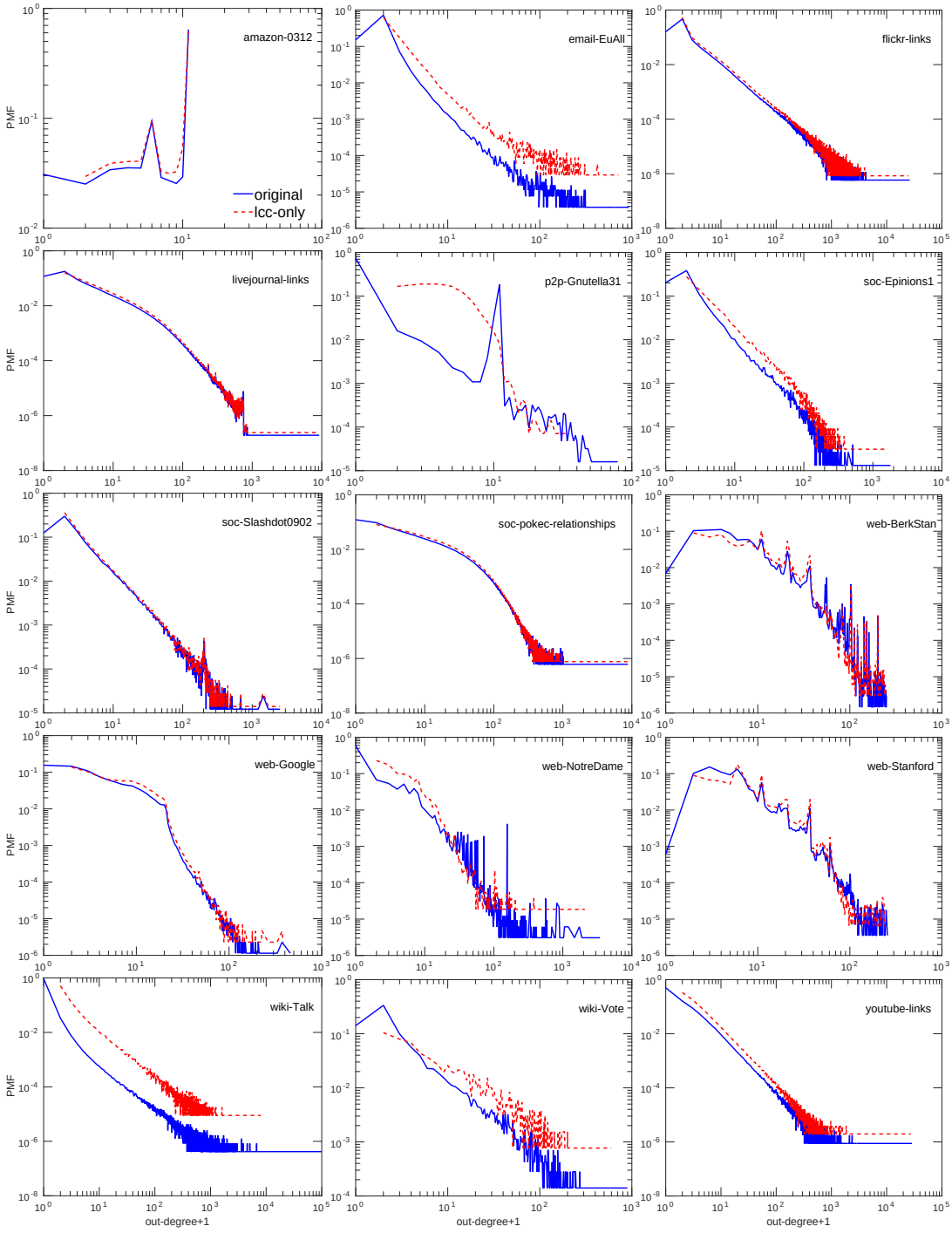


Figure 3.4. Out-degree probability mass function (p.m.f.) for each network and its largest strongly connected component (LCC). A large difference between these p.m.f.s suggests it is beneficial to use multiple walkers and/or random jumps.

For a given value of b, the number of uniform vertex samples is $B/(c+b)$. For each of the remaining $B - B/(c+b)$ samples, a vertex v is sampled in proportion to $\deg(v) + w$, where $\deg(v)$ is the undirected degree of v in $\mathcal{G}_u$. The choice of w and b impose, individually, a trade-off between estimation accuracy of the head and of tail of the distribution. For a fixed value of w, smaller values of b translate into better estimates of the head (and worse estimates of the tail) because we collect more (less) information about that region of the distribution from uniform vertex samples. For a fixed value of b, larger values of w also translate into more (less) accurate estimates of the head (tail), because random jumps are more likely to move a node to low in- and out-degree nodes (as they tend to occur more frequently).

In what follows, we observe through simulations that *despite the uniform edge sampling approximation, the previous intuition holds for H-DUFS head estimates, but not always for tail estimates*. In many cases, as we increase the number of walkers (i.e., decrease b) or increase w, we still obtain good estimates of the tail. This occurs because varying w or b changes the transition probability matrix that governs the sampling process, and thus, the sample distribution.

We simulate DUFS on each original network dataset for combinations of random jump weight $w \in \{0.1, 1, 10\}$ and budget per walker $b \in \{1, 10, 10^2, 10^3\}$ (1000 runs each). Values of w much smaller and much larger than these would be approximately equivalent to H-DUFS without jumps and random vertex sampling, respectively. Larger values of b would approximately correspond to DURW. We consider four scenarios that correspond to whether the incoming edges are directly observable or not and to two different costs of independent vertex sampling $c = 1$ or $c = 10$. Evaluating these parameter combinations is useful to establish practical guidelines for choosing H-DUFS parameters, which we summarize in Table 3.1. We observe that the estimation accuracy is somewhat monotonic w.r.t. the variation of each of these parameters, individually. This suggests that combinations other than the ones investigated here will not provide large accuracy gains.

Table 3.1. Practical guidelines on setting H-DUFS parameters to obtain accurate head or tail estimates depending on in-edge visibility and vertex sampling cost c.

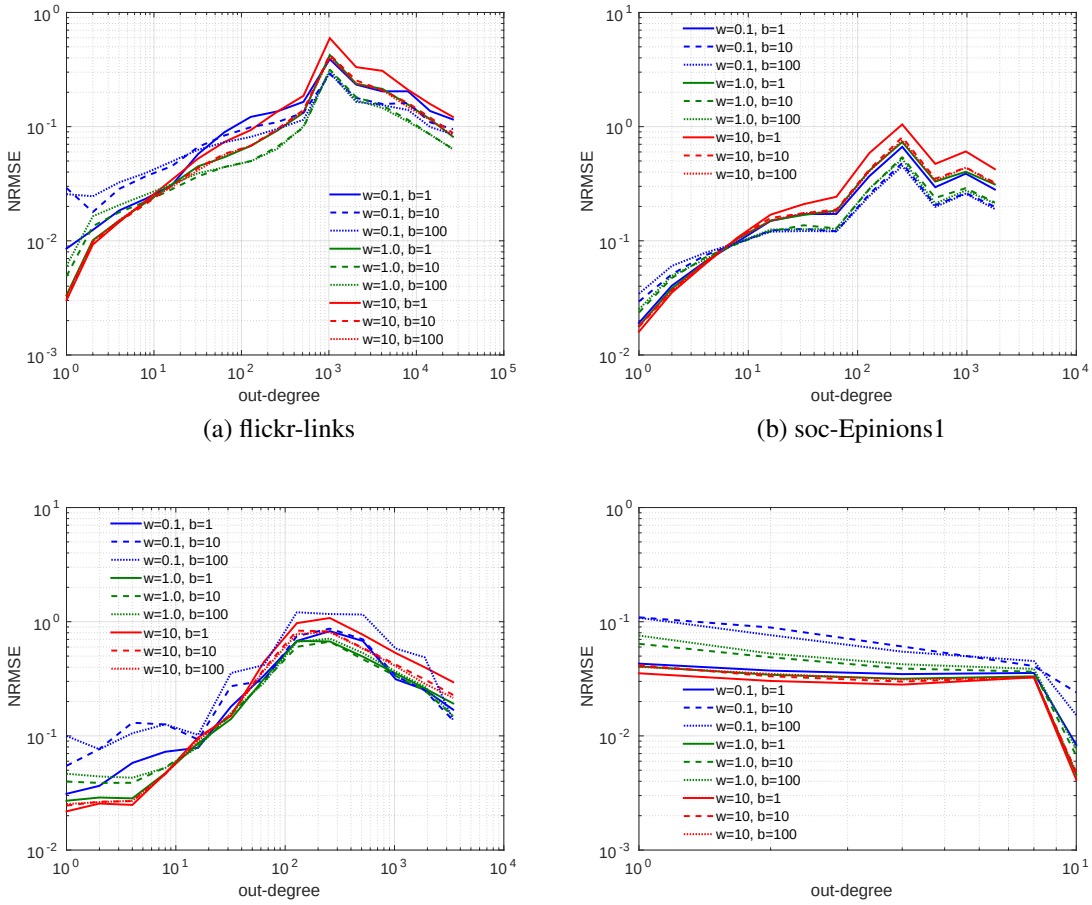
	uniform vertex sampling cost			
	$c = 1$		$c = 10$	
in-edges	visible	not visible	visible	not visible
most accurate for small out-degrees	$w = 10$ $b = 1$	$w = 10$ $b = 1$	$w = 1$ $b = 10^2$	$w = 10$ $b = 1$
most accurate for large out-degrees	$w = 1$ $b = 10$	$w = 1$ $b = 10, 10^2, 10^3$	$w = 0.1$ $b = 10^3$	$w = 0.1$ $b = 10, 10^2, 10^3$

3.5.1.1 Visible in-edges, $c = 1$

Figures 3.5(a-c) show typical results when varying w and b. To avoid clutter, we show only estimates for powers of two (or the closest out-degree values) and omit results for $b = 10^3$. Figure 3.5(d) shows similar results for amazon-0312, the dataset with the smallest maximum out-degree (max. is 10). Similar to our intuition for uniform edge sampling, the NRMSE associated with the head increases with b and decreases with w, on virtually all datasets.[4] Also as expected, for a fixed values of w, $b = 1$ yields larger errors in the tail than $b \in \{10, 100\}$ (except for amazon-0312). However, contrary to the intuition for uniform edge sampling, $w = 1$ matches or outperforms $w = 0.1$ for (except for $b = 1$). This is best visualized in Figure 3.5(d). This happens because setting $w = 1$ allows DUFS to sample regions with large probability mass (in this case, the head) and, at the same time, allows the sampler to move walkers from low volume to high volume components more often than $w = 0.1$. We also observe that $b = 10$ outperforms $b \in \{10^2, 10^3\}$ for $w \in \{0.1, 1\}$. Dataset amazon-0312 is the only dataset where $(w = 10, b = 1)$ obtained the best results over the entire out-degree distribution.

We note that the curves shown in Figures 3.5(a-d) seem to be composed of two distinct parts. In the beginning, the NRMSE increases with the out-degree and, after a given out-degree, the NRMSE starts to decrease. This behavior is investigated in Section 3.6.1.

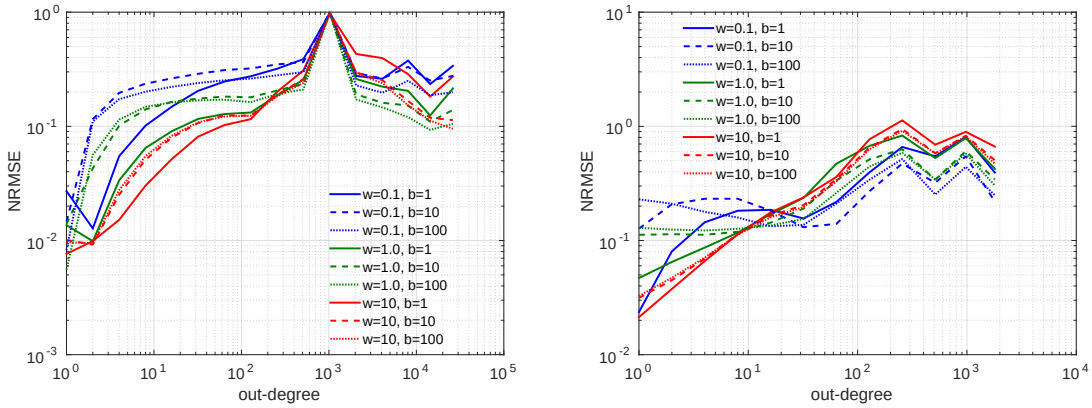
[4]For simplicity, the observations regarding the distribution head (tail) are based on the single smallest (largest) out-degree on each dataset. Similar conclusions are obtained when combining NRMSEs associated with several of the smallest (largest) out-degrees.



(a) flickr-links (b) soc-Epinions1

(c) web-NotreDame (d) amazon-0312

Figure 3.5. (visible in-edges) Effect of DUFS parameters on datasets with many connected components, when $B = 0.1|\mathcal{V}|$ and $c = 1$. Legend shows the average budget per walker (b) and jump weight (w). Trade-off shows that configurations that result in many random vertex samples, such as $(w = 10, b = 1)$, yield accurate head estimates, whereas configurations such as $(w = 1, b = 10)$ yield accurate tail estimates. Since NRMSE range varies across datasets, the y-axis limits are defined on a per-dataset basis.



(a) flickr-links (b) soc-Epinions1

Figure 3.6. (invisible in-edges) Effect of DUFS parameters on datasets with many connected components, when $B = 0.1|\mathcal{V}|$ and $c = 1$. Legend shows the average budget per walker (b) and jump weight (w). Configurations that result in many walkers which jump too often, such as $(w \geq 10, b = 1)$ yield accurate head estimates, whereas configurations such as $(w = 1, b = 10^3)$, yield accurate tail estimates.

3.5.1.2 Invisible in-edges, $c = 1$

The results we obtained are similar to those obtained for the visible in-edge scenario, but NRMSEs tend to be larger. Figures 3.6(a,b) show typical results for different DUFS parameters, represented by two datasets (also shown in the previous figure). Once again, the intuition for uniform edge sampling holds for the distribution head: decreasing b and increasing w yield more accurate estimates for the smallest out-degrees. While $b = 1$ results in poor estimates for the largest out-degrees, our intuition regarding w does not hold true for the tail. More precisely, in most cases $w = 1$ outperforms $w = 0.1$ (one exception being dataset soc-Epinions1). As opposed to the visible in-edge scenario, increasing b tends to provide more accurate tail estimates for $w = 1$. We investigate this effect in Section 3.5.3. We find that, for a fixed w, larger values of b make the random walks jump more often, moving them from small volume components to large volume components, yielding better tail estimates.

3.5.1.3 Visible in-edges, $c = 10$

Consider the case where the cost of obtaining uniform vertex samples is ten times larger than the cost of moving a walker. It is no longer clear that using many walkers and frequent random jumps achieves the most accurate head estimates, as this could rapidly deplete the budget. In fact, we observe that setting $w = 10$ or $b = 1$ yields poor estimates for both the smallest and largest out-degrees. While increasing the jump weight w or decreasing b sometimes improves estimates in the head, it rarely does so in the tail. The best results for the smallest out-degrees are often observed when setting $w = 1$ and $b = 10$ or 10^2. On the other hand, setting $(w = 0.1, b = 10^3)$ or $(w = 1, b = 10^2)$ usually achieves relatively small NRMSEs for the largest out-degree estimates.

3.5.1.4 Invisible in-edges, $c = 10$

Unlike the scenario with visible in-edges, setting $w = 10$ and $b = 1$ often produces the most accurate estimates for the smallest out-degrees. This is because many of the datasets have nodes with no out-edges; these nodes can only be reached through a neighbor or through uniform vertex sampling. Conversely, the general trend for tail estimates is similar to those observed for the visible in-edges case: large values of w and small values of b yield less accurate estimates for the largest out-degree values. For $w = 1$, however, $b = 10^2$ often outperforms $b = 10^3$. On the other hand, for $w = 0.1$ there is little difference in the estimates for different values of b.

3.5.2 Evaluation of DUFS in the visible in-edges scenario

In this section we compare two variants of Directed Unbiased Frontier Sampling: E-DUFS, which uses the edge-based estimator and H-DUFS, which uses the hybrid estimator, against a single random walk (SingleRW) and multiple independent random walks (MultiRW).

3.5.2.1 Out-degree and in-degree distribution estimates

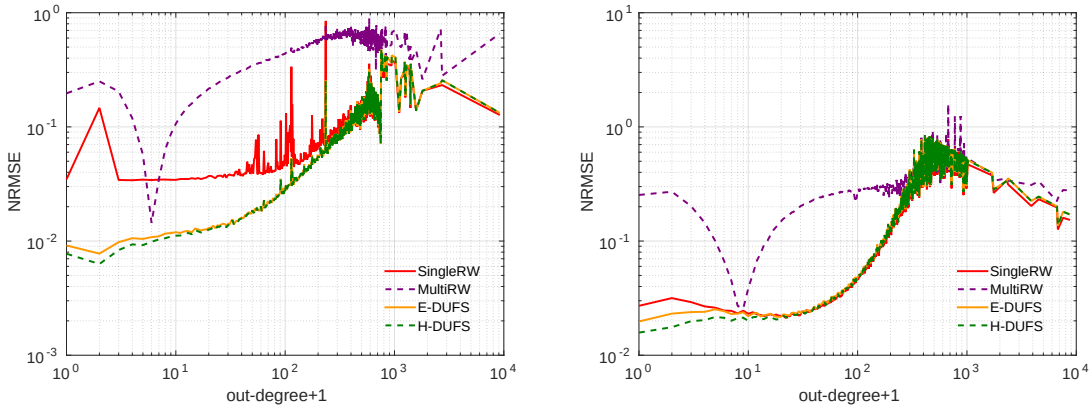
Here we focus on estimating the marginal in- and out-degree distributions. Each simulation consists of 1000 runs used to compute the empirical NRMSE. For MultiRW, E-DUFS and H-DUFS we set the average budget per walker to be $b = 10$. For conciseness, we only show a few representative results.

Figure 3.7 shows typical results obtained when using SingleRW, MultiRW, E-DUFS and H-DUFS to estimate out-degree distributions on the datasets. In eight out of 15 datasets, MultiRW yields much larger NRMSEs than does the SingleRW. As pointed out in [77, Section 4.5], this is due to the fact that the estimator in (3.2) assumes that all edges are sampled with the same probability. This assumption is violated by MultiRW because the stationary sampling probability depends on the size of the connected component within which each walker is located. E-DUFS estimates are consistently more accurate than those of MultiRW and SingleRW, except on datasets where the original graph and its LCC have similar out-degree distributions. In some of these cases SingleRW slightly outperforms E-DUFS in the tail (see Fig. 3.7(b)). H-DUFS, in turn, outperforms E-DUFS in the head of the out-degree distribution and has similar performance when estimating other out-degree values. For this reason, defining the estimation task in terms of the CCDF would give H-DUFS an unfair advantage.

When restricted to the largest connected component, the performance differences between SingleRW and E-DUFS and those between SingleRW and H-DUFS become smaller, for $B = 0.1|\mathcal{V}|$. Results for in-degree distribution estimation are qualitatively similar and are omitted.

3.5.2.2 Joint in- and out-degree distributions

We compare the NRMSEs associated with H-DUFS and SingleRW for the estimates of the joint in- and out-degree distribution. We observe that H-DUFS consistently outperforms SingleRW on all datasets. On 10 out of 15 datasets, the estimates corresponding to low in-



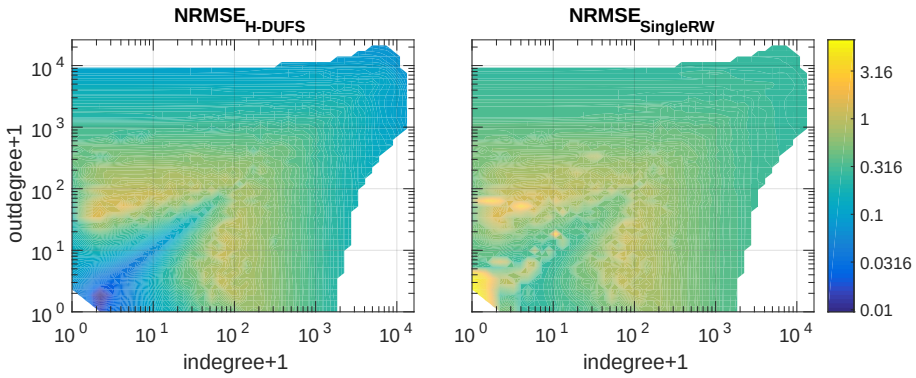
(a) livejournal-links

(b) soc-pokec-relationships

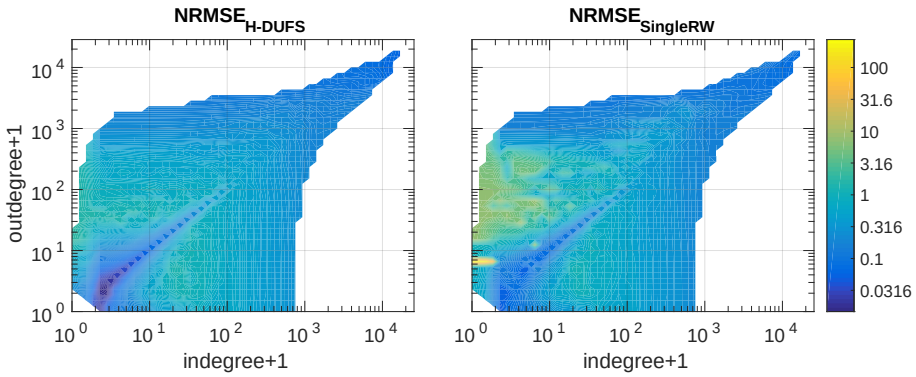
(c) web-BerkStan

(d) wiki-Talk

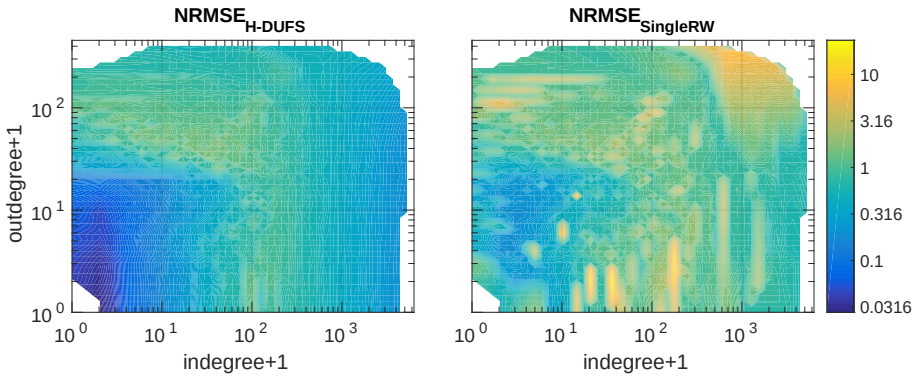
Figure 3.7. Comparison of single random walk (SingleRW), multiple independent random walks (MultiRW), DUFS with edge-based estimator (E-DUFS) and with hybrid estimator (H-DUFS). MultiRW yields the worst results, as the edge sampling probability is not the same across different connected components. Both DUFS variants outperform SingleRW, but H-DUFS is slightly more accurate in the head.



(a) flickr-links



(b) youtube-links



(c) web-Google

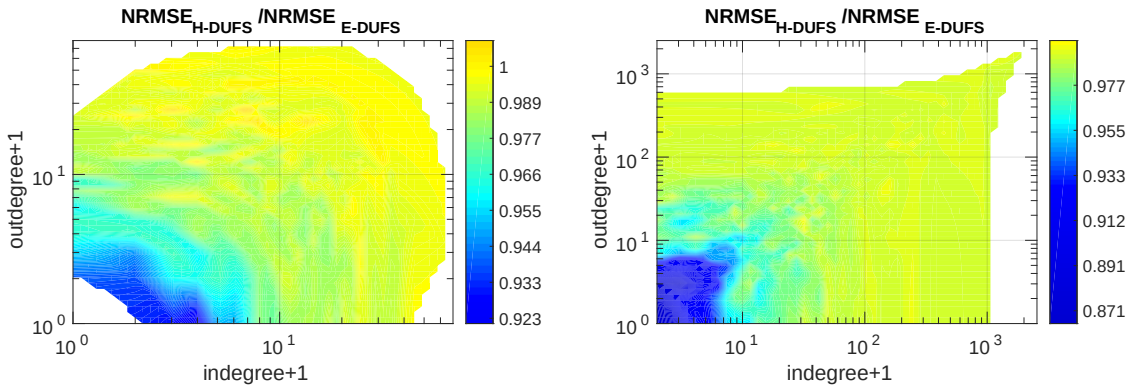
Figure 3.8. Comparison between H-DUFS and SingleRW w.r.t. NRMSE when estimating the joint in- and out-degree distribution. In most cases SingleRW will exhibit "hot spots" (regions with large NRMSE), which are mitigated by H-DUFS.

degree and low out-degree exhibit much smaller errors when using H-DUFS than when using SingleRW. Furthermore, H-DUFS also achieves smaller estimation errors for most of the remaining points of the joint distribution in 11 out of 15 datasets. Figures 3.8(a-b) show heatmaps corresponding to typical NRMSE results for H-DUFS and SingleRW. Interestingly, we note that on the web graph datasets and on the email-EuAll dataset, H-DUFS outperforms SingleRW by one or two orders of magnitude, as illustrated by Figure 3.8(c), which shows the heatmap comparison for dataset web-Google. Although the NRMSE exhibited by SingleRW applied to the LCC datasets is much smaller, the comparison between H-DUFS and SingleRW is qualitatively similar and is, therefore, omitted.

We next investigate performance gains obtained by using the hybrid estimator instead of the original estimator. Figures 3.9(a-b) show the ratios between the NRMSEs obtained with H-DUFS (hybrid) to those obtained with the E-DUFS (original) for two networks. We use the NRMSE ratio (or equivalently, the root MSE ratio) to make it easier to visualize the differences. We observe that H-DUFS consistently outperforms E-DUFS on all datasets. More precisely, the error ratio is rarely above one and, for points corresponding to small in- and out-degrees, it often lies below 0.9. Results on most datasets are similar to that depicted in Figure 3.9(a), but results on social networks datasets are closer to that shown in Figure 3.9(b), where large in- and out-degrees also seem to benefit from the information contained in the walkers' initial locations. Results for the LCC datasets are qualitatively similar, with accuracy gains from the hybrid estimator slightly larger on these datasets than on the original datasets.

3.5.3 Evaluation of DUFS in the invisible in-edges scenario

In this section, we compare the NRMSEs associated with DUFS and DURW when estimating out-degree distributions in the case where in-edges are not directly observable. We note that DURW is known to outperform a reference method for this scenario proposed in [10]. For a comparison between DURW and this reference method, please refer to [76].

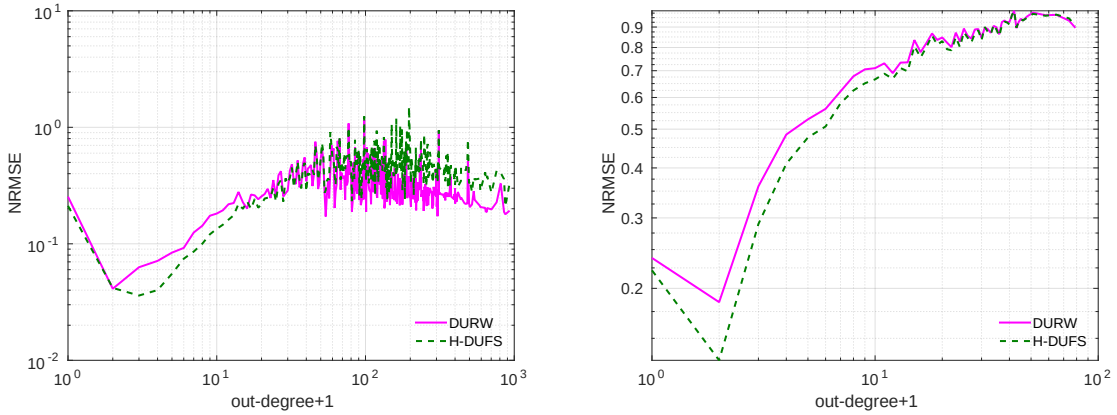


(a) p2p-Gnutella31 (b) soc-Slashdot0902

Figure 3.9. NRMSE ratios between H-DUFS and E-DUFS of the estimated joint in- and out-degree distribution for two datasets. H-DUFS is typically better than H-DUFS at low in and out-degree regions (**left**), but in social network graphs presented improvements over most of the joint distribution (**right**).

As we mentioned in Section 3.5.1, DURW results are similar to those obtained with DUFS when the budget per walker b is large, since DURW is a special case of DUFS where $b = B - c$. Therefore, we compare DURW and DUFS when b is small and the total number of uniform vertex samples collected by each method is roughly the same. More precisely, we simulate DUFS for $b = 10$ and $w = 1$ and set the DURW parameter w so that the number of vertex samples differs by at most 1% (averaged over 1000 runs). This aims to provide a fair comparison between these methods.

We find that neither of the two methods consistently outperforms the other over all datasets. The extra random jumps performed by DURW prevent the walker from spending much of the budget in small volume components. As a result, DURW tends to exhibit larger errors in the head but smaller errors in the tail of the out-degree distribution than DUFS. Figures 3.10(a-b) show typical results for $w = 1$ and $b = 10$. DUFS exhibited lower estimation errors in the head of the distribution on 11 datasets, being outperformed by DURW on one dataset and displaying comparable performance on the others. In six out of 15 datasets, DURW had better performance in the tail, while DUFS yielded better



(a) email-EuAll (b) p2p-Gnutella31

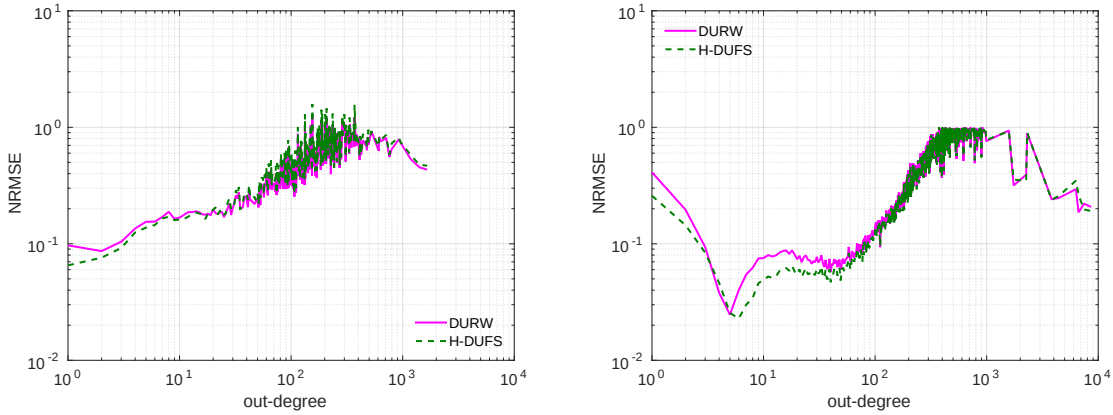
Figure 3.10. NRMSEs associated with DUFS ($b = 10$, $w = 1$) and DURW (w' chosen to match average number of vertex samples) when estimating out-degree distribution. DURW performs more random jumps, thus better avoiding small volume components. This improves DURW results in the tail, but often results in lower accuracy in the head (**left**). In one third of the datasets, DUFS yielded similar or better results than DURW over most out-degree points (**right**).

results on other five datasets. Results for $w = 1$ and $b \in \{10^2, 10^3\}$ are similar and are, therefore, omitted. As b increases, differences between DUFS and DURW start to vanish.

To better understand the impact of multiple connected components in DUFS and DURW performances, we simulate each method on the largest strongly connected component of each dataset (i.e., on the LCC datasets). Figures 3.11(a-d) show typical results among the LCC datasets. In most networks, DUFS yields smaller NRMSEs than DURW in the head and similar NRMSEs in the tail. Once again, for a larger b the performances of DUFS and DURW become equivalent.

3.6 Discussion

In this section, we investigate the reason why NRMSE typically increases with out-degree (in-degree) up to a certain value and then starts to decrease, as observed in Section 3.5. We also discuss the stopping criterion for DUFS, which so far was assumed to be



(a) soc-Epinions1-lcc

(b) soc-pokec-relationships-lcc

(c) web-Stanford-lcc

(d) wiki-Vote-lcc

Figure 3.11. NRMSEs associated with DUFS ($b = 10$, $w = 1$) and DURW (w' chosen to match average number of vertex samples) when estimating out-degree distribution.

$B = 0.1|\mathcal{V}|$. Last, we discuss the performance of E-DUFS and H-DUFS when a mechanism for obtaining uniform vertex samples is not available.

3.6.1 Relationship between NRMSE and out-degree distribution

In Section 3.5 we observed that NRMSE tends to increase with out-degree up to a certain point and to decrease after that. Moreover, for some out-degree ranges the NRMSE seems to vary linearly on the out-degree when plotted in log-log scale (see, for instance, Figure 3.5). For simplicity, we discuss the undirected graph case, but the extension to directed graphs is straightforward. We assume that each sampled edge results in exactly one observation, obtained by retrieving the set of labels associated with one of the adjacent vertices chosen uniformly at random. To simplify the exposition, we start by analyzing uniform vertex sampling.

Let $\mathbb{S} = \{s_1, \ldots, s_B\}$ be the sequence of sampled vertices. For uniform vertex sampling, the probability of observing a given label ℓ in $\mathcal{L}(s_i)$ is θ_ℓ, for any $i = 1, \ldots, B$. The minimum variance unbiased estimator of θ_ℓ is

$$T_\ell(\mathbb{S}) = \frac{1}{B} \sum_{i=1}^{B} \mathbb{1}\{\ell \in \mathcal{L}(s_i)\}. \tag{3.9}$$

Note that the summation in (3.9) is a binomially distributed random variable with parameters B and θ_ℓ. It follows that the mean squared error (MSE) of $T_\ell(\mathbb{S})$ is given by

$$\begin{aligned} \text{MSE}(T_\ell(\mathbb{S})) &= E[(T_\ell(\mathbb{S}) - \theta_\ell)^2] \\ &= \frac{\theta_\ell(1-\theta_\ell)}{B}. \end{aligned} \tag{3.10}$$

For uniform edge sampling, the probability of observing a given label $\ell \in \mathcal{L}$ in $\mathcal{L}(s_i)$ for $i = 1, \ldots, B$, is equal to

$$\pi_\ell = \frac{\sum_{v \in \mathcal{V}} \mathbb{1}\{\ell \in \mathcal{L}(v)\} \deg(v)}{\sum_{u \in \mathcal{V}} \deg(u)}.$$

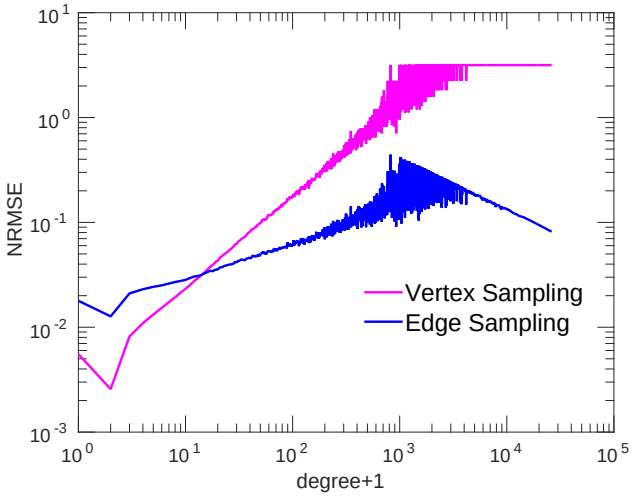


Figure 3.12. NRMSE from Uniform Vertex Sampling and Uniform Edge Sampling when estimating degree distribution on the Flickr dataset (for $B = 0.1|\mathcal{V}|$).

In that case, the following estimator can be shown to be unbiased

$$T'_\ell(\mathbb{S}) = \frac{1}{B}\sum_{k=1}^{B} \frac{\mathbb{1}\{\ell \in \mathcal{L}(s_k)\}\deg^{-1}(s_k)}{\sum_{j=1}^{B}\deg^{-1}(s_j)}. \tag{3.11}$$

In particular, when vertex labels are the undirected degrees of each node, the probability of observing a given label $\ell \in \mathcal{L}$ becomes $\pi_\ell = \ell\theta_\ell/d$, where d is the average undirected degree. The estimator for $B = 1$ reduces to $T'_\ell(\mathcal{S}_1) = \mathbb{1}\{\ell \in \mathcal{L}(s_1)\}$, which is a random variable distributed according to a Bernoulli with parameter π_ℓ. As a result, the MSE for $B > 1$ independent samples is given by

$$\text{MSE}(T'_\ell(\mathbb{S})) = \frac{\pi_\ell(1-\pi_\ell)}{B} \tag{3.12}$$

$$= \frac{\ell\theta_\ell(d-\ell\theta_\ell)}{d^2 B}. \tag{3.13}$$

Equations (3.10) and (3.12) characterize the conditions under which each sampling model is more accurate. More precisely, for all i such that $\theta_\ell > \pi_\ell$ (or equivalently, $\ell < d$), uniform vertex sampling yields better estimates than uniform edge sampling. This dichotomy is illustrated in Figure 3.12, which shows the NRMSE associated with degree

distribution estimates resulting from each sampling model on the flickr-links dataset [53], for $B = 0.1|\mathcal{V}|$.

Note that in log-log scale, both curves resemble a straight line for $\ell = 2, \ldots, 10^3$, which indicates a power law. For degrees larger than 5×10^3, the NRMSE associated with vertex sampling is constant, while the NRMSE associated with edge sampling decreases linearly on the degree. We show that these observations are direct consequences of the fact that the degree distribution in this network (as well as many other real networks) approximately follows a power law distribution. However, the degree distribution of a finite network cannot be an exact power law distribution because of the resolution limit. As a result, most of the largest degree values are observed exactly once. This can be seen in Figure 3.4 by noticing that on the flickr-links (and many other datasets) the p.m.f. is constant for the largest out-degrees. Assume, for instance, that the degree distribution can be modeled as

$$\theta_\ell = \begin{cases} \ell^{-\alpha}/Z\,, & 1 \leq \ell \leq \tau \\ 1/|\mathcal{V}|\,, & \ell > \tau, \end{cases}$$

for some $\alpha \geq 1$ and some normalizing constant Z.

From (3.10), we have that for uniform vertex sampling,

$$\text{NRMSE}(T_\ell(\mathbb{S})) = \sqrt{(1/\theta_\ell - 1)/B}. \tag{3.14}$$

For $\theta_l \ll 1$ (true for large degrees), this implies

$$\text{NRMSE}(T_\ell(\mathbb{S})) \approx \begin{cases} \sqrt{Z\ell^\alpha/B}\,, & 1 \leq \ell \leq \tau \\ \sqrt{|\mathcal{V}|/B}\,, & \ell > \tau. \end{cases} \tag{3.15}$$

Clearly, for $\ell > \tau$, the NRMSE is constant. For $1 \leq \ell \leq \tau$, taking the log on both sides yields

$$\log(\text{NRMSE}(T_\ell(\mathbb{S}))) \approx \frac{\alpha}{2}\log\ell + \frac{1}{2}(\log Z - \log B),$$

which explains the linear relationship on that range observed in Fig. 3.12.

From (3.12), we have that for uniform edge sampling,

$$\text{NRMSE}(T'_\ell(\mathbb{S})) = \sqrt{(1/\pi_\ell - 1)/B}. \tag{3.16}$$

For $\theta_l \ll 1$ (true for large degrees), this implies

$$\text{NRMSE}(T'_\ell(\mathbb{S})) \approx \begin{cases} \sqrt{Zd\ell^{\alpha-1}/B}\,, & 1 \leq \ell \leq \tau \\ \sqrt{\frac{|\mathcal{E}|}{\ell}/B}\,, & \ell > \tau. \end{cases} \tag{3.17}$$

Taking the log on both sides, it follows that

$$\log(\text{NRMSE}(T'_\ell(\mathbb{S}))) \approx \begin{cases} \frac{\alpha-1}{2}\log\ell + \frac{1}{2}(\log Z + \log d - \log B)\,, & 1 \leq \ell \leq \tau \\ -\frac{1}{2}\left(\log\ell - \log|\mathcal{E}| - \log B\right)\,, & \ell > \tau, \end{cases}$$

which explains the linear increase followed by the linear decrease observed in Fig. 3.12.

3.6.2 The stopping criterion

In all simulations described in this chapter, we set the budget to be $B = 0.1|\mathcal{V}|$. We did not study the effect of varying the budget because statistical theory gives us good intuition on how the error should vary with the number of samples (see 3.6.1). In practice, however, to set the budget to some fraction of the number of nodes, we have to estimate $|\mathcal{V}|$.

The *Random Tour* and the *Sample and Collide* methods proposed in [59] use RWs to estimate the number of nodes $|\mathcal{V}|$. In particular, to obtain an estimate with relative variance ϵ, *Sample and Collide* requires $O\left(\bar{d}\log(|\mathcal{V}|)\frac{\sqrt{|\mathcal{V}|/\epsilon}}{\lambda_2}\right)$ RW steps, where $\bar{d}$ is the average degree and λ_2 is the spectral gap of a RW on the underlying graph. While $|\mathcal{V}|$ and λ_2 are not known a priori, the authors describe a heuristic that can be used for deciding when to stop the sampling procedure.

Alternatively, instead of setting a sampling budget, a practitioner can define a stopping criterion based on the estimates obtained at a given step. In its simplest form, a distance measure (e.g., Euclidean distance) can be used to determine if the estimates have converged. A more sophisticated criterion would involve reasoning about the estimation error. For instance, assuming that the estimates obtained at a given step are the true distribution, it is possible to use the Cramér-Rao Bound to numerically compute a lower bound on the estimation error associated with E-DUFS or H-DUFS given the number of uniform vertex samples and random walk samples.

3.6.3 Performance of DUFS in the absence of uniform vertex sampling

In this section, we investigate the estimation accuracy of {E,H}-DUFS when the random walkers are *not* initialized uniformly over $\mathcal{V}$. We consider two simple non-uniform distributions over $\mathcal{V}$ to determine the initial walker locations walker positions:

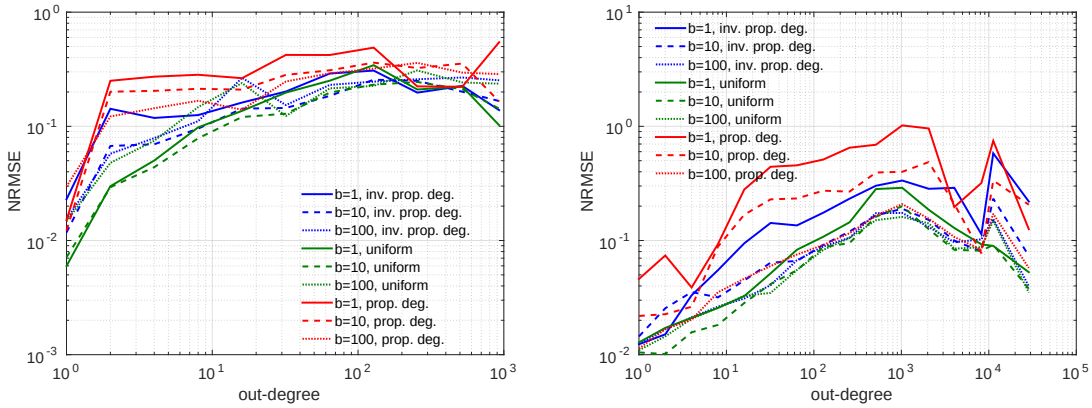
- Distribution PROP: proportional to the undirected degree, that is,

$$P(\text{initial walker location is } v) = \frac{\deg(v)}{\sum_{u\in\mathcal{V}} \deg(u)}; \tag{3.18}$$

- Distribution INV: proportional to the reciprocal of the undirected degree, that is,

$$P(\text{initial walker location is } v) = \frac{\deg^{-1}(v)}{\sum_{u\in\mathcal{V}} \deg^{1}(u)}. \tag{3.19}$$

We simulate E-DUFS and H-DUFS on each network dataset setting the budget per walker to $b \in \{1, 10, 10^2\}$ (100 runs), under the scenario where in-edges are visible. Since we assume uniform vertex sampling (VS) is not available, we must set the random jump weight to $w = 0$. We include, though, results obtained when the initial walker locations are determined via VS for comparison. Figures 3.13(a,b) shows typical results in terms of the NRMSE associated with E-DUFS out-degree distribution estimates. We observe that NRMSE decreases with the budget per walker until $b = 10^2$, both for PROP and INV.

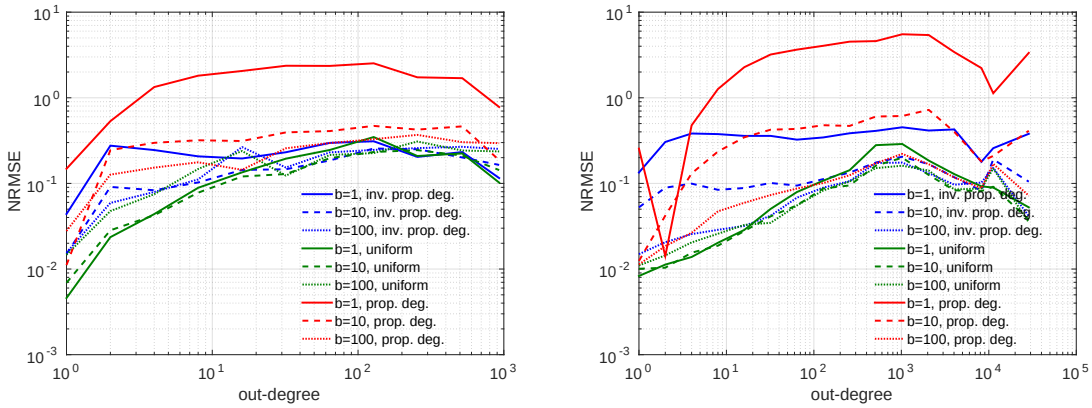


(a) email-EuAll (b) youtube-links

Figure 3.13. Effect of initializing walkers non-uniformly over $\mathcal{V}$ on E-DUFS accuracy. NRMSE decreases with budget per walker until $b = 10^2$.

Although it is clear that using the hybrid estimator when the initial walker locations come from some non-uniform distribution can incur unknown – and potentially large – biases, we include similar results obtained with H-DUFS for completeness. Intuitively, since H-DUFS assumes that observations from initial walker locations come from VS, the estimation errors tend to grow with the number of walkers n. Figures 3.14(a,b) shows the NRMSE associated with H-DUFS out-degree distribution estimates for the same datasets as before. As expected, errors are larger for smaller values of b (or equivalently, large n). In particular, we observe that the NRMSE associated with large out-degrees is larger when sampling according to INV than when sampling according to PROP. This occurs because the former tends to overrepresent large out-degrees, which are usually associated with very small probability masses. On the other hand, INV tends to overrepresent small out-degrees, slightly underestimating the mass associated with large out-degrees.

In summary, the previous results indicate that when the initial walker locations are determined according to some unknown distribution, a practitioner should use E-DUFS with large b (e.g., 10^2). We conduct additional simulations setting $b = B - 1$ (i.e., using a



(a) email-EuAll (b) youtube-links

Figure 3.14. Effect of initializing walkers non-uniformly over $\mathcal{V}$ on H-DUFS accuracy. NRMSE associated with H-DUFS is generally larger than that associated with E-DUFS. NRMSE decreases rapidly in b. Errors associated with large out-degrees are especially high when walkers are more likely to start on large degree nodes (distribution PROP).

single walker), which yielded poor results and are, therefore, omitted. H-DUFS yields very inaccurate estimates if their initial positions are not chosen via VS.

3.7 Related Work

Crawling methods for exploring undirected graphs: A number of papers investigate crawling methods (e.g., breadth-first search, random walks, etc.) for generating subgraphs with similar topological properties as the underlying network [39, 52]. On the other hand, [58] empirically investigates the performance of such methods w.r.t. specific measures of representativeness that can be useful in the context of specific applications (e.g., finding high-degree nodes for outbreak detection). However, these works focus on techniques that yield biased samples of the network and do not possess any accuracy guarantees. [2, 47] demonstrate that Breadth-First-Search (BFS) introduces a large bias towards high degree nodes, and it is difficult to remove these biases in general, although it can be reduced if the network in question is almost random [47]. Random walk (RW) is biased to sample high degree nodes, however its bias is known and can be easily corrected [77].

Random walks in the form of Respondent Driven Sampling (RDS) [34, 81] have been used to estimate population densities using snowball samples of sociological studies. The Metropolis-Hasting RW (MHRW) [86] modifies the RW procedure, aimed at sampling nodes with equal probability to estimation errors introduced by sampling. [18, 72] analytically prove that MHRW degree distribution estimates perform poorly in comparison to RWs. Empirically, the accuracy of RW and MHRW has been compared in [29, 71] and, as predicted by the theoretical results, RW is consistently more accurate than MHRW.

Reducing the mixing time of a regular RW is one way of improving the performance of RW based crawling methods. [9] proves that random jumps increase the spectral gap of the random walk, which in turn, leads to faster convergence to the steady state distribution. [46] assigns weights to nodes that are computed using their neighborhood information, and develop a weighted RW-based method to perform stratified sampling on social networks. They conduct experiments on Facebook and show that their stratified sampling technique achieves higher estimation accuracy than other methods. However, the neighborhood information in their method is limited to helping find random walk weights and is not used in estimators of graph statistics of interest. To solve this problem, [19] randomly samples nodes (either uniformly or with a known bias) and then uses neighborhood information to improve its unbiased estimator. [94] modifies the regular random walk by "rewiring" the network of interest on-the-fly in order to reduce the mixing time of the walk.

Crawling methods for exploring directed graphs: Estimating observable characteristics by sampling a directed graph (in this case, the Web graph) has been the subject of [10] and [36], which transform the directed graph of web-links into an undirected graph by adding reverse links, and then use a MHRW to sample webpages uniformly. Our "backward edge traversal" is an adaptation of the method of [10] to work with a pure random walk and random jumps. Both of these Metropolis-Hastings RWs are designed to sample directed graphs and do not allow random jumps. However, the ability to perform random jumps (even if jumps are rare) makes DURW and DUFS more efficient and accurate than

the MetropolisHastings RW algorithm. Random walks with PageRank-style jumps are used in [52] to sample large graphs. In [52], however, there is no technique to remove the large biases induced by the random walk and the random jumps, which makes this method unfit for estimation purposes. More recently, another method based on PageRank was proposed in [80], but it assumes that obtaining uniform vertex samples is not feasible. In the presence of multiple strongly connected components, this method offers no accuracy guarantees.

Graph sketching: In the last decade, there has been a growing interest in graph sketching for processing massive networks. A sketch is a compact representation of data. Unlike a sample, a sketch is computed over the entire graph, that is observed as a data stream. For a survey on graph sketching techniques, please refer to [61].

3.8 Conclusion

In this chapter, we proposed the Directed Unbiased Frontier Sampling (DUFS) method for characterizing networks. DUFS generalizes the Frontier Sampling (FS) and the Directed Unbiased Random Walk (DURW) methods. In some sense, DUFS extends FS to make it applicable to directed networks when incoming edges are not directly observable by building on ideas from DURW. Like DURW, DUFS can also be applied to undirected networks without any modification.

We also proposed a new estimator for vertex label distribution that can account for FS and DUFS walkers initial locations – or more generally, uniform vertex samples – and a heuristic that can reduce the variance incurred by vertex samples that happen to sample nodes whose labels have extremely low probability masses. When the proposed estimator is used in combination with the heuristic, we showed that estimation errors can be significantly reduced in the distribution head when compared with the estimator proposed in [77], regardless of whether we are estimating out-degree, in-degree or joint in- and out-degree distributions.

We conducted an empirical study on the impact of DUFS parameters (namely, budget per walker and random jump weight) on the estimation of out-degree and in-degree distributions using a large variety of datasets. We considered four scenarios, corresponding to whether incoming edges are directly observable or not and whether random vertex sampling has a similar or larger cost than moving random walkers on the graph. This study allowed us to provide practical guidelines on setting DUFS parameters to obtain accurate head estimates or accurate tail estimates. When the goal is a balance between the two objectives, intermediate configurations can be chosen.

Last, we compared DUFS against random walk-based methods designed for undirected and directed networks. In our simulations for the scenario where in-edges are visible, DUFS yielded much lower estimation errors than a single random walk or multiple independent random walks. We also observed that DUFS consistently outperforms FS due to the degree proportional jumps mechanism implemented by the former. In the scenario where in-edges are unobservable, DUFS outperformed DURW when estimating the probability mass associated with the smallest out-degree values (for equivalent parameter settings). In addition, more often than not, DUFS slightly outperformed DURW when estimating the mass associated to the largest out-degrees. In the presence of multiple strongly connected components, DURW tends to move from small to largest components more often than DUFS, sometimes exhibiting lower estimation errors in the distribution tail. However, when restricting the estimation to the largest component, DUFS outperforms DURW in virtually all datasets used in our simulations. These and other results showed in this chapter indicate that random jumps are not an alternative to the use of multiple walkers, but rather a complementar mechanism that can further improve performance of random walk-based techniques.

CHAPTER 4

ESTIMATION OF SET-SIZE DISTRIBUTION AND CHARACTERIZATION OF LARGE NETWORKS VIA SAMPLING

4.1 Introduction

Networks are increasingly large and complex; they pose tremendous challenges to their characterization in the wild. Characterizing network structure (e.g. degree distribution), network traffic flows (e.g. TCP/IP flow sizes in communication networks), node labels (e.g. group memberships), is usually impossible without resorting to sampling due to the size and scale of current networks. Practitioners often sample networks in order to characterize them. One important way of characterizing networks from samples is to estimate their out-degree and in-degree distributions. Estimating in-degree distributions is especially challenging because incoming edges to a node v are often not directly observable (e.g., links to a given web page). These incoming edges can be seen as elements of a set represented by node v. A process that samples network edges is then a process that reveals some of the elements belonging to different sets. Like estimating in-degree distributions, many problems in network characterization through sampling can be mapped into the problem of estimating the set-size distribution (SSD). The SSD estimation problem can be stated as follows. Consider a collection of non-overlapping sets whose elements are probabilistically sampled. The problem is to estimate the underlying set-size distribution based on the samples.

SSD estimation has several applications. As described earlier, one application of particular interest is the estimation of in-degree distributions of on-line social networks, where nodes represent people and a directed edge represents, for instance, one or more messages

exchanged between two pairs of nodes. By monitoring message exchanges over a period of time one samples a fraction of the edges. In this case, the user relationships on the network correspond to elements of a given set, in the SSD estimation problem. Using these samples we want to estimate the in-degree or out-degree distribution of nodes. The SSD problem also manifests itself in estimating the distribution (in packets) of TCP/UDP flow sizes [21]. In flow size estimation, each packet is probed (sampled) with a fixed probability. Each packet is associated with a TCP/UDP flow. The goal is to estimate the distribution of flow sizes from probed packets.

Despite the importance of characterizing set-size distributions, to the best of our knowledge no deep analysis of SSD estimation exists in the literature. We fill this gap and *prove the existence of a phase transition on the estimation accuracy of in-degree distributions of arbitrarily large power-law graphs* (more precisely, any heavier-than-exponential distribution). Namely, if less than 50% of the edges are observed that for *any estimator* (be it frequentist or Bayesian), the lower bound on estimation errors grows with the network size. Moreover, when we only observe nodes with at least one edge sampled, even a first order metric like average degree is subject to the same threshold behavior, i.e., sampling less than 50% of all incoming edges impedes the estimation of in-degree averages. As a result, in the SSD estimation problem an increase in the number of samples may, paradoxically, result in no increase in accuracy. We prove these and other results in the general setting of sets with arbitrary set-size distribution.

4.1.1 General Observations

We uncover some properties of the set-size distribution (SSD) estimation, including:

- *A (finite) increase in samples can result in* no *reduction in estimation errors.*

Unlike estimation problems such as election polls, where a sufficient increase in samples always results in increased accuracy, we show, paradoxically, that in the SSD estimation

problem an increase in samples can result in no increase in accuracy. Section 4.4 unveils the cause of this behavior and explains how to avoid it. Another interesting property is:

- *In networks with power-law set-size distributions (our results hold for any heavier-than-exponential distributions), randomly sampling less than 50% of the set elements (e.g., edges of a node) provides* almost no *information about the set-size distribution or the average set-size. On the other hand, accurate SSD estimation is always possible in networks with sub-exponential set-size distributions.*

The above observation is interesting because power-law distributions have more tail probability mass and, thus, large sets are more likely to have more sampled elements than when the distributions have sub-exponential tails. However, and despite this, we show that if less than 50% of elements are sampled, then estimates of power-laws distributions (more precisely, any heavier-than-exponential distribution) are significantly less accurate than the estimates obtained from sub-exponential distributions. We also prove the existence of a similar phase transition for exponential distributions, but the corresponding threshold depends on the distribution exponent. Our work also provides a host of other puzzling observations, fully and formally presented in Section 4.4.

4.1.2 Outline

This chapter is organized as follows. In Section 4.2 we conduct experiments on the indegree distribution estimation with real data. Section 4.3 presents the sampling and estimation models. Sections 4.4 and 4.5 present our theoretic results. Section 4.6 discusses problems of interest to field analysts, highlighting common mistakes made in the literature and how to avoid them. In Section 4.7, we discuss the related work. Finally Section 4.8 presents the conclusions and outlook.

4.2 Estimation with Real Data

In this section, we investigate through simulation one particular application of the set-size distribution problem: the estimation of the in-degree distribution of a network. Consider the Enron email dataset [44], that describes a network composed by a group of people who exchanged emails during a certain period of time. Here each node represents a person and two people have a directed edge if one has emailed the other. The maximum node in-degree is 1383.

Collecting a fraction of the exchanged messages means sampling network edges. Suppose that each directed edge is observed independently with probability p, regardless of the number of messages collected over an edge. Henceforth, the number of observed incoming edges to a node, provided that this number is at least one, will be called a sample. Figure 4.1(a) depicts the quality of the maximum likelihood estimator for node in-degree with $p = 0.25$, leading to $N = 10^4$ sampled individuals. The black dots indicate the true in-degree distribution, the blue curve shows a typical estimate of that distribution, and the heat map indicates the density of estimated values across 100 runs, where red indicates high density and yellow (white) indicates low (no) density of estimated values. We observe from the blue curve that the estimated values can be orders of magnitude away from the actual values and from the heat map we observe that the blue line is typical.

In what follows we illustrate the effects of varying the number of samples N or changing the sample probability p separately. To vary N while keeping p fixed, we draw a node in-degree directly from the in-degree distribution of Enron email network and subsequently sample its edges. We repeat this process until we obtain N observed sets. This can be seen as sampling a larger (smaller) network that has the same degree distribution.

We make two main observations:

1. **Increasing the number of samples does not reduce estimation error.** This is an odd behavior. We know from estimation theory that the error should decrease by a factor of $\sqrt{k}$ when the number of samples is increased by a factor of k. Figure 4.1(b)

shows the corresponding results for $N = 50 \times 10^3$. We observe that the estimated fraction of nodes of each degree still varies from the actual values.

To make it clear that the accuracy gain from increasing the number of samples is not in agreement with theory, we compute the estimation error obtained when we vary the number of samples $N \in \{5, 10, 20, 50, 100\} \times 10^3$, for $p = 0.25$. The error is measured in terms of the Normalized Root Mean Square Error (NRMSE).Then we take the average NRMSE from the head (degrees up to 10) and the tail (degrees larger than 10) of the distribution separately.

Surprisingly, we observe in Figure 4.1(c) that there is almost no improvement in accuracy across different sample sizes, even when we compare 5×10^3 and 10^5 samples. We also display in this figure the expected reduction in the NRMSE for both head and tail by dashed lines. It turns out that the error does not decrease as we might expect. This raises the question of why, which we address in Section 4.4.

2. **For much larger values of p, the error starts to decrease with the number of samples.** According to Theorem 4.1 presented in Section 4.4, the difficulties experienced above arise due to the use of small sampling probability ($p < 0.5$) with heavy-tailed distributions, and not due to a lack of samples. Hence we repeat the experiment using $p = 0.9$. Figures 4.1(d) and 4.1(e) show the heat maps for $N = 20 \times 10^3$ and $N = 10^5$. As opposed to what we previously saw, increasing the number of samples does increase the accuracy of the estimate. The accuracy gain as a function of the number of samples is shown in Figure 4.1(f). In fact, we observe that the NRMSE does decrease as expected for the head of the distribution, but not for the tail. Why are there two distinct behaviors, one for the head and one for the tail? Why did it help to increase the number of samples when estimating frequencies of small degrees for $p = 0.9$, as opposed to what we observed for $p = 0.25$? Is it possible to make the NRMSE of the tail to decrease as fast as the NRMSE of the head?

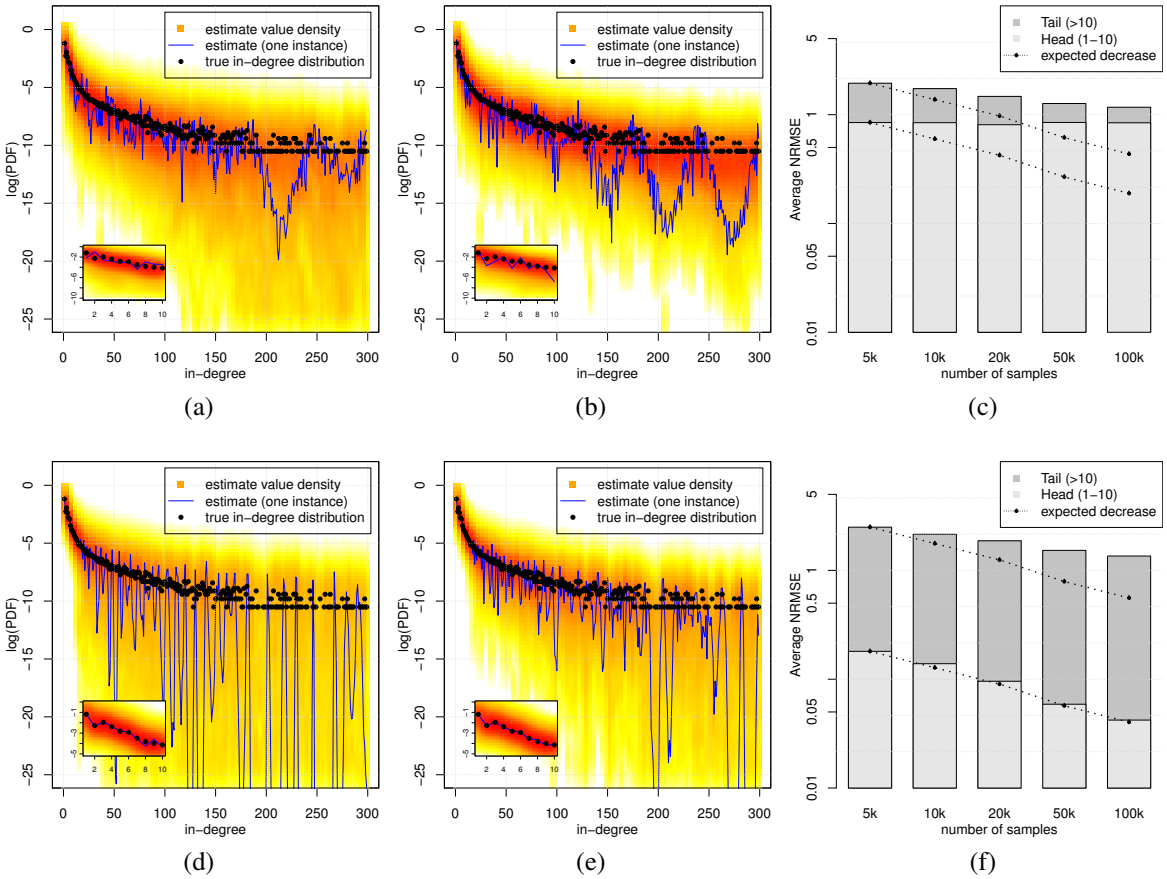


Figure 4.1. The first row (a-c) shows the results for $p = 0.25$, while the second row (d-e) shows the corresponding plots for $p = 0.90$. (a-b,d-e) True degree distribution, one example of estimate and heat map indicating the occurrence rates of the estimate values for $N = 10 \times 10^3$ samples (first column) and $N = 50 \times 10^3$ samples (second column), respectively. The red color in the heat map indicates high density of estimated values and yellow (white) indicates low (no) density of estimated values. A subplot shows a zoom-in for the first degrees. (c,f) Average NRMSE of the head and the tail of the distribution for $N \in \{1, 5, 10, 20, 100\} \times 10^3$. Dashed line shows how the error should vary with the number of samples. In (c) we have the **typical behavior of wrong estimates**. Increasing the number of samples does not improve the quality of estimates. On the other hand (f) shows the **typical behavior of correct estimates**. Here increasing the number of samples yields lower estimation errors of the head.

In order to investigate the questions we pose here, we compute the Cramér-Rao Lower Bound (CRLB) of the SSD estimation problem. This give us a lower bound on the estimation errors based on the amount of information contained in the samples, measured in terms of Fisher Information. Moreover, we apply the CRLB to the estimation of the in-degree distribution and average in-degree.

4.3 Model

Let $\mathcal{S}_k$ be a nonempty set of elements, $k = 1, \ldots, n$, with $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$, $i, j = 1, \ldots n$, $i \neq j$. Let $S_k = |\mathcal{S}_k|$ denote the size of the k-th set and assume set sizes are i.i.d. with distribution $S_k \sim \boldsymbol{\theta} = (\theta_1, \ldots, \theta_W)$, $W > 1$ $k \geq 1$.

We assume W is finite ($W < \infty$). The model divides elements into groups (sets) and our task is to characterize those groups from an incomplete observation (sample) of these groups. To illustrate the model, consider a directed graph; edges can be grouped by the nodes they are incident to (depart from), in which case $\mathcal{S}_k$ is the set of incoming (outgoing) edges of a node k, $\boldsymbol{\theta}$ is the indegree (outdegree) distribution, and W is the maximum indegree (outdegree). Another straightforward example is characterizing IP traffic in a communications network, where k is a TCP flow, $\mathcal{S}_k$ is the set of TCP/IP packets that constitute flow k, and W is the maximum observable flow size.

4.3.1 Sampling

We observe (sample) elements of $\mathcal{S}_k$, $k = 1, \ldots, n$, with probability p – a process also known as thinning. Let $\alpha(\mathcal{S}_k)$ be a random function that returns the number of observed elements of $\mathcal{S}_k$. Elements are sampled independently (i.e., the sampling process is Bernoulli) and thus,

$$P[\alpha(\mathcal{S}_k) = j | \, S_k = i] = \begin{cases} \binom{i}{j} p^j q^{i-j}, & 0 \leq j \leq i, i > 1 \\ 0, & \text{otherwise,} \end{cases}$$

where $q = 1 - p$. We assume that when no elements of a set are observed, then the set as a whole is not observed, i.e., $\mathcal{S}_k$ is said to be observable only if $\alpha(\mathcal{S}_k) > 0$. Thus, we denote

$$\mathbb{S} = \{\alpha(\mathcal{S}_k) : \alpha(\mathcal{S}_k) > 0\,,\, k = 1, \ldots, n\}$$

the set of the observable set-sizes. Let $N = |\mathbb{S}|$ denote the number of observed sets.

4.3.2 Estimation

We start by considering $p = 1$, that is, all elements of all sets are observed. The minimum variance estimator of θ_i is

$$T'_i(\mathcal{S}_1, \ldots, \mathcal{S}_n) = \sum_{k=1}^{n} \frac{\mathbb{1}\{S_k = i\}}{N},$$

where $N = n$. To measure the accuracy of the estimates we consider the mean squared error (MSE) – a.k.a. quadratic loss – of the estimates

$$\begin{aligned}\text{MSE}(T'_i(\mathcal{S}_1, \ldots, \mathcal{S}_n)) &= E[(T'_i(\mathcal{S}_1, \ldots, \mathcal{S}_n) - \theta_i)^2] \\ &= \frac{\theta_i(1-\theta_i)}{n} \leq \frac{1}{4n}.\end{aligned}$$

Thus, for $p = 1$ the estimation error decreases as $1/n$, recalling that n is the number of sets.

Unfortunately, accurately estimating $\boldsymbol{\theta}$ when $p < 1$ is significantly more challenging. Recall that a set $\mathcal{S}_k$ is said to be observable if $\alpha(\mathcal{S}_k) > 0$. *We assume that unobservable sets cannot be used in the estimation process.* That is, our estimator only has access to sets $\mathcal{S}_k$ where $\alpha(\mathcal{S}_k) > 0$. Here we need another function T_i that takes the observed set-sizes $\mathbb{S}$ as inputs and outputs an **unbiased** estimate $T_i(\mathbb{S})$ of θ_i, i.e., $E[T_i(\mathbb{S})] = \theta_i$. In what follows we focus on unbiased estimates. The function T_i that minimizes the MSE with respect to sets of size $i = 1, \ldots, W$ is

$$T_i^\star(\mathbb{S}) = \arg\min_{T_i} E[(T_i(\mathbb{S}) - \theta_i)^2],\ i = 1, \ldots, W \qquad \text{s.t. } E[T_i^\star(\mathbb{S})] = \theta_i.$$

4.4 Results

In order to investigate the questions we pose here, we study the Cramér-Rao Lower Bound (CRLB) of the problem of estimating the set-size distribution. This gives us a lower bound on the mean square error based on the amount of information contained in the samples, measured in terms of Fisher Information. Moreover, we apply the CRLB to the estimation of the in-degree distribution and average in-degree. The proofs can be found in Appendices B to F.

We now introduce our main results which derive MSE lower bounds for unbiased set-size distribution and average set-size estimators. In addition, we show that the lower bounds obtained for the set-size distribution are achievable by a Maximum Likelihood Estimator. We consider a general formulation of the sampling problem, where the number of observed sets N is constant and N is independent of the maximum degree W. We also consider the sampling probability p to be a known constant.

Theorem 4.1. *Let $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_W)$ be the set-size distribution, $\mathbb{S}$ be the sequence of N observed set-sizes after randomly sampling elements of the sets with probability p, and $T_i(\mathbb{S})$, $i \geq 1$ be an unbiased estimator of θ_i.*

1. *If θ_W decreases faster than exponentially in W, i.e., $-\log \theta_W = \omega(W)$, then MSE$(T_i(\mathbb{S})) = \Omega(1/N)$, provided $0 < p < 1$.*

2. *If θ_W decreases exponentially in W, i.e., $-\log \theta_W = W \log a + o(W)$ for some $0 < a < 1$, then*

   (a) $\log[\text{MSE}(T_i(\mathbb{S}))] = \Omega(W - \log N)$, *provided* $p < a/(a+1)$,

   (b) $\text{MSE}(T_i(\mathbb{S})) = \Omega(W^{2i+1}/N)$, *provided* $p = a/(a+1)$,

   (c) $\text{MSE}(T_i(\mathbb{S})) = \Omega(1/N)$, *provided* $p > a/(a+1)$.

3. *If θ_W decreases more slowly than exponential, i.e., $-\log \theta_W = o(W)$, then*

   (a) $\log[\text{MSE}(T_i(\mathbb{S}))] = \Omega(W - \log N)$, *provided* $p < 1/2$,

(b) **MSE**$(T_i(\mathbb{S})) = \omega(1/N)$, *provided* $p = 1/2$ *and* $\sum_{j=1}^{W} j^{2i}\theta_j = \omega(1)$,

(c) **MSE**$(T_i(\mathbb{S})) = \Omega(1/N)$, *provided either* $p > 1/2$, *or* $p = 1/2$ *and* $\sum_{j=1}^{W} j^{2i}\theta_j = O(1)$.

The lower bounds of type $\Omega(1/N)$ in Theorem 4.1 are only meaningful if they are achievable. We investigate this achievability question in Appendix E, showing that, in fact, there exists a Maximum Likelihood Estimator (MLE) $T_i^\star(\mathbb{S})$ of θ_i, $i = 1, \ldots, W$ that is asymptotically efficient and normal, which means that $T_i^\star(\mathbb{S})$ approaches the CRLB uniformly as $N \to \infty$. Hence, the corresponding bounds for $T_i^\star(\mathbb{S})$ are as follows.

Theorem 4.2. *Let* $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_W)$ *be the set-size distribution,* $\mathbb{S}$ *be the sequence of observed set-sizes after randomly sampling elements of the sets with probability* p*, and* $T_i^\star(\mathbb{S})$, $i \geq 1$ *is the MLE of* θ_i *when* $N \to \infty$.

1. *If* θ_W *decreases faster than exponentially in* W*, i.e.,* $-\log \theta_W = \omega(W)$*, then* **MSE**$(T_i^\star(\mathbb{S})) = \Theta(1/N)$*, provided* $0 < p < 1$.

2. *If* θ_W *decreases exponentially in* W*, i.e.,* $-\log \theta_W = W \log a + o(W)$ *for some* $0 < a < 1$ *and* $p > a/(a+1)$*, then* **MSE**$(T_i^\star(\mathbb{S})) = \Theta(1/N)$.

3. *If* θ_W *decreases more slowly than exponential, i.e.,* $-\log \theta_W = o(W)$ *and* $p \geq 1/2$*, then* **MSE**$(T_i^\star(\mathbb{S})) = \Theta(1/N)$.

In what follows we consider the problem of estimating the average set-size $m_{\boldsymbol{\theta}} = \sum_{i=1}^{W} i\theta_i$ from the sample $\mathbb{S}$. Surprisingly, we obtain bounds analogous to the bounds for the set-size distribution in Theorem 4.1. This result is surprising because the average observed set-size $m_\phi = \sum_{i=1}^{W} i\phi_i$ has remarkably different bounds: m_ϕ is always finite (independent of p or W) as long as the second moment of ϕ is finite (see Section 4.5).

Theorem 4.3. *Let* $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_W)$ *be the set-size distribution,* $\mathbb{S}$ *be the sequence of* N *observed set-sizes after randomly sampling elements of the sets with probability* p*, and* $\hat{m}_{\boldsymbol{\theta}}(\mathbb{S})$ *be an unbiased estimator of* $m_{\boldsymbol{\theta}}$.

1. *If θ_W decreases faster than exponentially in W, i.e., $-\log \theta_W = \omega(W)$, then MSE$(\hat{m}_{\boldsymbol{\theta}}(\mathbb{S})) = \Omega(1/N)$, provided $0 < p < 1$.*

2. *If θ_W decreases exponentially in W, i.e., $-\log \theta_W = W \log a + o(W)$ for some $0 < a < 1$, then*

   (a) $\log[$*MSE*$(\hat{m}_{\boldsymbol{\theta}}(\mathbb{S}))] = \Omega(W - \log N)$, *provided* $p < a/(a+1)$,

   (b) *MSE*$(\hat{m}_{\boldsymbol{\theta}}(\mathbb{S})) = \Omega(W/N)$, *provided* $p = a/(a+1)$,

   (c) *MSE*$(\hat{m}_{\boldsymbol{\theta}}(\mathbb{S})) = \Omega(1/N)$, *provided* $p > a/(a+1)$.

3. *If θ_W decreases more slowly than exponential, i.e., $-\log \theta_W = o(W)$, then*

   (a) $\log[$*MSE*$(\hat{m}_{\boldsymbol{\theta}}(\mathbb{S}))] = \Omega(W - \log N)$, *provided* $p < 1/2$,

   (b) *MSE*$(\hat{m}_{\boldsymbol{\theta}}(\mathbb{S})) = \omega(1/N)$, *provided* $p = 1/2$ *and* $\sum_{j=1}^{W} j^2\theta_j = \omega(1)$,

   (c) *MSE*$(\hat{m}_{\boldsymbol{\theta}}(\mathbb{S})) = \Omega(1/N)$, *provided either* $p > 1/2$ *or* $p = 1/2$ *and* $\sum_{j=1}^{W} j^2\theta_j = O(1)$.

From Theorem 4.3 we see that estimating the average set-size is asymptotically as hard as estimating the distribution $\boldsymbol{\theta}$. We conjecture that the analogue of Theorem 4.2 for the average set-size can be found by showing that the conditions for the existence of the MLE also hold for $m_\theta(\mathbb{S})$.

In what follows we sketch the proofs of Theorems 4.1 and 4.3 and describe their implications. The full proofs can be found in the Appendices B to F.

4.4.1 Lower Bound on Estimation Errors

In this section we derive a lower bound on the Mean Squared Error (MSE) of $T_i(\mathbb{S})$, $i = 1, \ldots, W$. For this we use the Cramér-Rao (CR) lower bound of $T_i(\mathbb{S})$, which gives the smallest MSE that any unbiased estimator T_i can achieve.

Recall that a set is observable only if one or more of its elements are observable. The probability that a (random) set $\mathcal{S}$ has j observed elements, given $j > 0$, is defined as

$$b_{ji}(p) \equiv P\left[\alpha(\mathcal{S}) = j \,|\, \alpha(\mathcal{S}) > 0, |\mathcal{S}| = i\right] = \frac{\binom{i}{j} p^j q^{i-j}}{1 - q^i}, \tag{4.1}$$

if $0 < j \leq i \leq W$ and $b_{ji}(p) = 0$ otherwise, where $q = 1 - p$ and $\alpha(\mathcal{S}) > 0$ is the size of $\mathcal{S}$ after sampling. Let $d_j(\boldsymbol{\theta}, p)$ denote the fraction of observed sets with exactly j observed elements. From (4.1) we have, $j = 1, \ldots, W$,

$$\begin{aligned} d_j(\boldsymbol{\theta}, p) &= P[\alpha(\mathcal{S}) = j | \, |\mathcal{S}| > 0] \\ &= \sum_{i=j}^{W} P[\alpha(\mathcal{S}) = j | \alpha(\mathcal{S}) > 0, |\mathcal{S}| = i] \times P[|\mathcal{S}| = i | \, \alpha(\mathcal{S}) > 0] \\ &= \sum_{i=j}^{W} b_{ji}(p) \phi_i(\boldsymbol{\theta}). \end{aligned} \tag{4.2}$$

where

$$\phi_i(\boldsymbol{\theta}) = P[|\mathcal{S}| = i \,|\, \alpha(\mathcal{S}) > 0] = \frac{\theta_i (1 - q^i)}{\sum_{k=1}^{W} \theta_k (1 - q^k)}, \tag{4.3}$$

is the distribution of the observed set-sizes. Or, in matrix notation,

$$d(\boldsymbol{\theta}, p) = B(p) \boldsymbol{\phi}(\boldsymbol{\theta}),$$

where $d(\boldsymbol{\theta}, p) = (d_1(\boldsymbol{\theta}, p), \ldots, d_W(\boldsymbol{\theta}, p))^{\mathsf{T}}$ and $B(p) = [b_{ji}(p)], j, i = 1, \ldots, W$. To illustrate the distribution $d(\boldsymbol{\theta}, p)$ in our model, note that for a random observed set $\mathcal{S}$,

$$\alpha(\mathcal{S}) \sim d(\boldsymbol{\theta}, p),$$

with likelihood function

$$P[\alpha(\mathcal{S}) = j \,|\, \boldsymbol{\theta}] = (B(p) \boldsymbol{\phi}(\boldsymbol{\theta}))_j = d_j(\boldsymbol{\theta}, p), \quad j = 1, \ldots, W. \tag{4.4}$$

In what follows for simplicity we denote $d_j(\boldsymbol{\theta}, p)$ by $d_j(\boldsymbol{\theta})$.

We now apply the Cramér-Rao Theorem to find the lower bounds of MSE$(T_i(\mathbb{S}))$. The Cramér-Rao Theorem states that the MSE of any unbiased estimator T is lower bounded by the inverse of the Fisher information matrix divided by the number of independent samples N, provided some weak regularity conditions hold [89, Chapter 2], i.e.,

$$\text{MSE}(T_i(\mathbb{S})) \equiv E[(T_i(\mathbb{S}) - \theta_i)^2] \geq \frac{\left((J^{(\boldsymbol{\theta})}(p))^{-1}\right)_{ii}}{N}, 1 \leq i \leq W. \tag{4.5}$$

where $(J^{(\boldsymbol{\theta})}(p))^{-1}$ is the inverse of the Fisher information matrix of a *single* set-size observation defined using the likelihood function (4.4) as

$$(J^{(\boldsymbol{\theta})}(p))_{i,k} \equiv \sum_{j=1}^{W} \frac{\partial d_j(\boldsymbol{\phi}(\boldsymbol{\theta}))}{\partial \theta_i} \frac{\partial d_j(\boldsymbol{\phi}(\boldsymbol{\theta}))}{\partial \theta_k} \frac{1}{d_j(\boldsymbol{\phi}(\boldsymbol{\theta}))}, \tag{4.6}$$

given $\sum_{i=1}^{W} \theta_i = 1$.

The lower bound in (4.5) is known in the literature as the Cramér-Rao lower bound or *CRLB* for short. Let $T_i(\mathbb{S})$ be an unbiased estimator, $i = 1, \ldots, W$. We say $T_i(\mathbb{S})$ is asymptotically efficient if MSE$(T_i(\mathbb{S}))$ approaches the Cramér-Rao lower bound in (4.5) as $N \to \infty$. We show in Appendix E that the Maximum Likelihood Estimator is asymptotically efficient on the set-size estimation under the condition that the bound is finite. In what follows we represent $J^{(\boldsymbol{\theta})}(p)$ as $J^{(\boldsymbol{\theta})}$ for simplicity.

4.4.2 Obtaining the CRLB

In what follows we derive the CRLB in closed-form as a function of the original set-size distribution $\boldsymbol{\theta}$, the sampling probability p, and the number of observed sets N, where we ignore the constraint $\sum_{i=1}^{W} \theta_i = 1$. Deriving a closed-form solution for the inverse of $J^{(\boldsymbol{\theta})}$ is no easy task as matrix $J^{(\boldsymbol{\theta})}$ is a function of $\partial \boldsymbol{\phi}(\boldsymbol{\theta}) / \partial \theta_i$, $j, i = 1, \ldots, W$, which makes

$J^{(\boldsymbol{\theta})}$ a non-linear function of $\boldsymbol{\theta}$. The Fisher information matrix in (4.6) can be derived as a function of $\boldsymbol{\phi}$ and thus

$$J_{i,k}^{(\boldsymbol{\phi})} \equiv \sum_{j=1}^{W} \frac{\partial d_j(\boldsymbol{\phi})}{\partial \phi_i} \frac{\partial d_j(\boldsymbol{\phi})}{\partial \phi_k} \frac{1}{d_j(\boldsymbol{\phi})}, \tag{4.7}$$

given $\sum_{i=1}^{W} \phi_i = 1$; and because $d_j(\boldsymbol{\phi})$ is linear in $\boldsymbol{\phi}$ the above yields

$$(J^{(\boldsymbol{\phi})})^{-1} = B(p)^{-1}\text{diag}(B(p)\boldsymbol{\phi})^{-1}(B(p)^{-1})^{\mathsf{T}} - \boldsymbol{\phi}\boldsymbol{\phi}^{\mathsf{T}}. \tag{4.8}$$

Here the term $\boldsymbol{\phi}\boldsymbol{\phi}^{\mathsf{T}}$ corresponds to the accuracy gain obtained by considering the constraint $\sum_{i=1}^{W} \phi_i = 1$ (see Tune and Veitch [88] for more details and Gorman and Hero [30] for the general formula on adding equality constraints to the CRLB). Quantitatively we can safely ignore the constant term $\boldsymbol{\phi}\boldsymbol{\phi}^{\mathsf{T}}$ as we are interested in the behavior of $(J^{(\boldsymbol{\phi})})^{-1}$ as a function of W and the elements of $\boldsymbol{\phi}\boldsymbol{\phi}^{\mathsf{T}}$ must be smaller than one. All that is left to do is to find a relationship between $(J^{(\boldsymbol{\phi})})^{-1}$ and $(J^{(\boldsymbol{\theta})})^{-1}$.

We now obtain $(J^{(\boldsymbol{\theta})})^{-1}$ from $(J^{(\boldsymbol{\phi})})^{-1}$ through the Jacobian $\nabla H = [h_{ik}]$, $h_{ik} = \partial\theta_i(\boldsymbol{\phi})/\partial\phi_k$ with $\theta_i(\boldsymbol{\phi})$ obtained from inversion of (4.3), we arrive at the equivalent multivariate rule [89, p. 83] and express $(J^{(\boldsymbol{\theta})})^{-1}$ as

$$(J^{(\boldsymbol{\theta})})^{-1} = \nabla H (J^{(\boldsymbol{\phi})})^{-1} \nabla H^{\mathsf{T}}. \tag{4.9}$$

Using (4.8) – detailed derivation relegated to Appendix B – we find:

$$[(J^{(\boldsymbol{\phi})})^{-1}]_{ij} = \sum_{k=\max(i,j)}^{W} \left(\frac{q}{p}\right)^{2k} \binom{k}{j}\binom{k}{i}(-1)^{-i-j}(q^{-i}-1) \times (q^{-j}-1)d_k(\boldsymbol{\theta}). \tag{4.10}$$

Substituting (4.10) into (4.9) – and applying a variety of algebraic manipulations detailed in Appendix G – yields

$$[(J^{(\boldsymbol{\theta})})^{-1}]_{ii} = \frac{1}{\eta^2}\Bigg(\underbrace{\frac{1}{(1-q^i)^2}[(J^{(\boldsymbol{\phi})})^{-1}]_{ii}}_{A_1(i)} + \underbrace{\theta_i^2 \sum_{j=1}^{W}\sum_{k=1}^{W} \frac{[(J^{(\boldsymbol{\phi})})^{-1}]_{kj}}{(1-q^k)(1-q^j)}}_{A_2(i)} - \underbrace{2\theta_i \sum_{j=1}^{W} \frac{[(J^{(\boldsymbol{\phi})})^{-1}]_{ij}}{(1-q^j)(1-q^i)}}_{A_3(i)} \Bigg), \tag{4.11}$$

where $\eta = \sum_{j=1}^{W} \phi_j(\boldsymbol{\theta})/(1-q^j)$. Note that term $A_1(i)$ of (4.11) is proportional to the CRLB of $\boldsymbol{\phi}$, $[(J^{(\boldsymbol{\phi})})^{-1}]_{ii}$ but terms $A_2(i)$ and $A_3(i)$ are more involved. Through a series of algebraic manipulations of terms A_1, A_2, and A_3, all detailed in Appendix B, we find that $(A_1(i) + A_2(i) - A_3(i))$ grows as a function of $(1-p)/p$ and W, yielding the relation

$$\text{MSE}(T_i(\mathbb{S})) = \Omega\left(\frac{\sum_{j=1}^{W} \left(\frac{1-p}{p}\right)^j \theta_j}{N} \right), \quad i = 1, \ldots, W, \tag{4.12}$$

where the number of observed sets N is large but constant with respect to W.

The result in (4.12) is very powerful as it gives a simple estimation error lower bound as a function of the sampling probability p and the original set-size distribution $\boldsymbol{\theta}$. In particular, the following examples applied to (4.12) give some intuition on the results in Theorem 4.1 – a detailed exposition is Appendix D. For instance, a close look at (4.12) reveals that when $((1-p)/p)^i\theta_i = \Omega(i^{-1})$ for all $i > i^\star$, $i^\star \ll W$, then the sum in (4.12) grows at least as fast as the harmonic series, which grows as $\log W$. On the other hand, when $((1-p)/p)^i\theta_i = O(i^{-\beta})$, $\beta > 1$, the sum in (4.12) converges to a constant, more precisely, it grows no faster than a Riemann zeta function with parameter β, $\zeta(\beta)$.

Thus, for a given $\boldsymbol{\theta}$ with $W \gg 1$ the CRLB exhibits an interesting sharp threshold (p_0) related to the sampling probability p. If $p < p_0$ no estimator T_i of θ_i, $i = 1, \ldots, W$, is able to achieve accurate estimates of θ_i. If $p > p_0$, there exist estimators $T_i(\mathbb{S})$, $i = 1, \ldots, W$ that can achieve accurate estimates, as $N \to \infty$. To be more specific, we look at the threshold behavior of p by breaking down $\boldsymbol{\theta}$ into three broad classes of distributions:

1. If θ_W decreases faster than exponentially in W there is no threshold behavior of p. This is because when $-\log \theta_W = \omega(W)$ there exists a constant $a < 1$ such that $((1-p)/p)^j \theta_j < a^j$, $j = 1, 2, \ldots$. Hence, the sum in (4.12) converges to a constant for any $p > 0$, yielding MSE$(T_i(\mathbb{S})) = \Omega(1/N)$, for $0 < p < 1$.

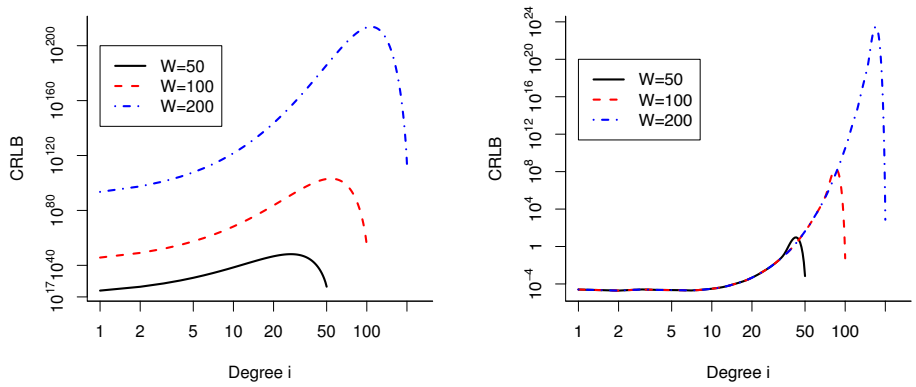
2. If $-\log \theta_W = W \log a + o(W)$ and $p \leq a/(a+1)$, then $((1-p)/p)^j \theta_j = a^{-j}\theta_j = \Omega(1)$, $\forall j$. Hence, the sum in (4.12) diverges with W. On the other hand, if $p > a/(a+1)$ the sum in (4.12) converges to a constant.

3. Finally, if θ_W decreases more slowly than exponential and $p < 1/2$, then there exists an $\epsilon > 0$, such that $((1-p)/p)^j > (1+\epsilon/2)^j$, $\forall j$. Because θ_j decreases more slowly than an exponential the sum in (4.12) diverges with W. If $p \geq 1/2$ the lower bound in (4.12) converges to a constant.

To illustrate our results, we compute the MSE lower bounds in (4.11) where $\boldsymbol{\theta}$ is the indegree distribution of the Enron email dataset truncated at different values of W. More precisely, we take the in-degree distribution of the Enron dataset (discussed in Section 4.2) and truncate the maximum degree to W by accumulating in W all the probability mass previously corresponding to degrees greater than W. The Enron in-degree distribution is a (truncated) heavier-than-exponential distribution.

Figures 4.2a and 4.2b show the MSE lower bounds for $p \in \{0.25, 0.90\}$, respectively. We observe that for $p = 0.25$ (Figure 4.2(a)) the MSE lower bound grows with W even for small degrees, as predicted by Theorem 4.1. While, for $p = 0.9$ (Figure 4.2(b)) the MSE lower bound behaves (mostly) independent of W, also as predicted by Theorem 4.1. These results corroborate to explain the simulations results in Section 4.2.

Other metrics besides the set-size distribution are of interest. In what follows we observe that the accuracy is similar to that of set-size distribution estimators T_i, $i = 1, \ldots, W$. We then analyze the accuracy of the average set-sizes.



(a) $p = 0.25$ (b) $p = 0.90$

Figure 4.2. CRLB of the in-degree distribution of the Enron dataset for $N = 10^4$ samples.

4.5 Accuracy of Estimated Averages

In this section we consider the accuracy of unbiased average set-size estimates. Let $m_{\boldsymbol{\theta}} = \sum_{j=1}^{W} j\theta_j$ be the average set-size. Theorem 4.3 implies that estimating the average set-size is in the same order of hardness as estimating the entire set-size distribution (see proof in Appendix F). However, we show that the average size of the observed sets, i.e., the average set-size in respect to $\boldsymbol{\phi}$, $m_{\boldsymbol{\phi}} = \sum_{j=1}^{W} j\phi_j$, is much easier to estimate.

Theorem 4.3 shows that estimating the average set-size is asymptotically as hard as estimating the distribution $\boldsymbol{\theta}$. However, the average size of the observed sets, i.e., the average set-size in respect to $\boldsymbol{\phi}$,

$$m_{\boldsymbol{\phi}} = \sum_{j=1}^{W} j\phi_j,$$

is much easier to estimate accurately. A estimation error bound for $m_{\boldsymbol{\phi}}$ is affected only by the first and second moments of $\boldsymbol{\phi}$, that is, as long as $m_{\boldsymbol{\phi}}$ and

$$m_{\boldsymbol{\phi}}^{(2)} = \sum_{j=1}^{W} j^2\phi_j$$

are finite, $m_{\boldsymbol{\phi}}$ can be accurately estimated if enough sets are observed. To prove this, let $\hat{m}_\phi(\mathbb{S})$ denote an unbiased estimate of m_ϕ and let

$$\text{MSE}(\hat{m}_\phi(\mathbb{S})) = E[(\hat{m}_\phi(\mathbb{S}) - m_\phi)^2]$$

denote the MSE of $\hat{m}_\phi(\mathbb{S})$. After applying a variety of algebraic manipulations detailed in Appendix F we arrive at the following inequality

$$\begin{aligned}
\text{MSE}(\hat{m}_\phi) &\geq \frac{(1,\ldots,W)(J^{(\phi)})^{-1}(1,\ldots,W)^{\mathrm{T}} - m_\phi^2}{N} \\
&= \sum_{k=1}^{W}\sum_{i=1}^{k}\sum_{j=1}^{k} ij\binom{k}{j}\binom{k}{i}\frac{(-q)^{2k-i-j}}{p^{2k}}(1-q^i)(1-q^j)d_k(\boldsymbol{\phi}) \\
&= \left(\sum_{i=1}^{W}\frac{i(pi+q^{i+1}-2q^i+q)\phi_i}{p(1-q^i)} - m_\phi^2\right)/N.
\end{aligned}$$

Interestingly,

$$\hat{m}_\phi^\star(\mathbb{S}) = \frac{\sum_{s\in\mathbb{S}} s}{Np} + \left(1-\frac{1}{p}\right)\frac{\sum_{s\in\mathbb{S}} \mathbb{1}\{s=1\}}{N}, \tag{4.13}$$

is an unbiased efficient (minimum variance) estimator of m_ϕ, yielding

$$\text{MSE}(\hat{m}_\phi^\star(\mathbb{S})) = \left(\sum_{i=1}^{W}\frac{i(pi+q^{i+1}-2q^i+q)\phi_i}{p(1-q^i)} - m_\phi^2\right)/N.$$

Alternatively we can rewrite the above as

$$\text{MSE}(\hat{m}_\phi^\star) = O\left(\frac{m_\phi^{(2)} - m_\phi^2}{N}\right).$$

Hence, $\text{MSE}(\hat{m}_\phi^\star)$ is lower bounded by the variance of the observed set-sizes. A simple explanation for this behavior is likely found in the inspection paradox. The sampling is biased towards sets with larger sizes, which increases the variance of the observed set-sizes and, in turn, makes it harder to unbias the samples.

4.6 Discussion

This section considers applications of our results to the estimation of Internet flow sizes on high-speed routers and to the estimation of latent indegrees (outdegrees) in directed networks (e.g. Web graph) through edge sampling, and bounds on Bayesian and biased estimators that complements our bounds of unbiased estimators. Moreover, we also discuss the practical aspects of the initialization of estimation procedures.

4.6.1 Application Example

An important problem in network traffic measurement and planning is to estimate the distribution of the flow sizes traversing a network. In this context, packets can be seen as elements grouped by flows. Due to efficiency requirements, packets traversing a link are sampled independently with a given probability, rather than collecting information about all packets. In the most common sampling design, two parameters are chosen: p, the sampling probability and N, the number of flows to be observed. A router is then set to sample packets traversing a link until N different flows are sampled, but it only actually stops after the last flow terminates. Here the number of observed sets (flows) is a constant defined a priori. More generally, this sampling design can be used in any scenario where a stream of elements is to be sampled. In the above scenario N and W are fixed and our results can be directly applied. There are alternative sampling scenarios, however, where N is a random variable.

4.6.2 Variable Number of Observed Sets (N)

Our results assume that the number of observed sets N, the maximum set-size W, and the sampling probability p are independent constants. In what follows we consider N to be a random variable that depends on the number of sets V and on the constants W and p, showing how our results can still be applied.

To exemplify this scenario, consider again the estimation of Internet flow-size distribution, where we sample packets with probability p from a fixed number of flows V.

In this case, N is binomially distributed random variable with parameters V and $\rho = \sum_{i=1}^{W} \theta_i(1-q^i)$ (probability of observing one set chosen uniformly at random). The Chernoff bound shows that N is concentrated around its mean $V\rho$:

$$P(N > (1+\delta)V\rho) \leq \exp\left(-\frac{V\rho\delta^2}{3}\right), \, 0 < \delta < 1.$$

For instance, choosing $\delta = \sqrt{\frac{3}{V\rho}\log\frac{1}{\epsilon}}$, yields $P(N > (1+\delta)V\rho) \leq \epsilon$. Even though V and ρ may be unknown in practice, this inequality still illustrates that, for a fixed ϵ, the upper bound on N increases linearly with V. From this fact, it follows that the negative implications (items 2(a) and 3(a)) of Theorems 4.1, 4.2, and 4.3 hold as long as W grows faster than $\log(V)$.

4.6.3 The Maximum Set Size W as a function of the Number of Sets V

Consider the following network sampling application. We wish to sample nodes on a directed network by randomly sampling edges (e.g. observing Web pages through sampling incoming links). Here a relationship between V and W arises, as the size of the network also determines the maximum node degree. To estimate the indegree distribution from these sampled edges we need to consider that W and V are coupled: the maximum indegree cannot exceed the number of nodes in the network. Nevertheless, on power-law networks, one can show that if W grows as $\Omega(V^b)$ with high probability, for any $b > 0$, then we can readily apply items 2(a) and 3(a) of Theorems 4.1 and 4.3. This is true, for instance, for Barabási-Albert networks, where W grows as $\Omega(V^{1/2})$ with high probability [63]. Since $-\log(\theta_W) \propto \log(W) = o(W)$ (and clearly the number of observed sets $N \leq V$), it follows from 3(a) that when $p < 1/2$, $\log[\text{MSE}(T_i(\mathbb{S}))] = \Omega(V^{1/2} - \log(V)) \to \infty$ as $V \to \infty$.

Also it is worth noting that all results listed in Theorems 4.1, 4.2, and 4.3 that do not depend on W also hold true, even when W and N depend on each other. For instance, for large Erdős-Renyi networks where the degree distribution asymptotically approaches either Poisson or Normal distributions, θ_W decreases faster than exponentially and our results

show that the MSE is lower bounded by $O(1/N)$ and if $N \to \infty$ then MLE achieves this bound.

4.6.4 Impact on Different Types of Estimators: Bayesian, Biased and Unbiased

To extend our results beyond unbiased estimators we explain the connection between Fisher information, the Cramér-Rao bound and biased estimators. We also extend our results to Bayesian estimators (including maximum a posteriori estimators).

4.6.4.1 Extension to Biased Estimators

Let $b(\theta_i) = E[T_i(\mathbb{S})] - \theta_i$ be the estimator bias. Then (see Ben-Haim and Eldar [12])

$$\text{MSE}(T_i(\mathbb{S})) \geq \left(1 + \frac{\partial b(\theta_i)}{\partial \theta_i}\right)^2 [(J^{(\boldsymbol{\theta})})^{-1}]_{ii},$$

assuming $\partial b(\theta_i)/\partial \theta_i$ exists. Note if the bias derivative satisfies $-2 < \partial b(\theta_i)/\partial \theta_i < 0$, then the biased estimator has a smaller MSE than any unbiased estimator. However, we believe it is unlikely that a biased estimator can be designed to compensate for a large value of $[(J^{(\boldsymbol{\theta})})^{-1}]_{ii}$ (as large as 10^{160} as seen in Section 4.4.2 for the Enron e-mail network).

4.6.4.2 Extension to Bayesian Estimators

Let $\boldsymbol{\theta}$ now be a random variable with prior distribution $\pi_{\boldsymbol{\theta}}$. A Bayesian estimator adds $\pi_{\boldsymbol{\theta}}$ as extra information to the estimation problem. The Fisher information of the prior is

$$J_{ij}^{(p)} = E\left[\frac{\partial \ln \pi_{\boldsymbol{\theta}}}{\partial \theta_i} \frac{\partial \ln \pi_{\boldsymbol{\theta}}}{\partial \theta_j}\right].$$

The Fisher information obtained exclusively by the data is $J^{(\boldsymbol{\theta})}$ presented in (4.6). And the total Fisher information *prior* + *data* is [89, p. 84]

$$J^{(t)} = J^{(p)} + J^{(\boldsymbol{\theta})}.$$

The Cramér-Rao bound of a Bayesian estimator $W_i(\mathbb{S})$ of θ_i with prior $\pi_{\boldsymbol{\theta}}$ yields [89, p. 85]

$$\text{MSE}(W_i(\mathbb{S})) \geq (J^{(t)})^{-1} = (J^{(p)} + J^{(\boldsymbol{\theta})})^{-1}.$$

Thus, if the data contains little Fisher information then any decrease in the MSE is due to the information contained in the prior $\pi_{\boldsymbol{\theta}}$.

4.6.5 Initialization of Estimation Procedures

As previously stated, eq. (4.4) can be used to derive a maximum likelihood estimator (MLE) for $\boldsymbol{\theta}$. From the MLE one could either use a constrained non-linear optimization method to maximize the likelihood function directly or use the Expectation-Maximization (EM) algorithm to write an iterative estimation procedure. In the latter case, the procedure consists of an initialization step followed by a loop of two steps known as the E-step and M-step. We discuss two issues that arise when EM is used to estimate the set-size distribution.

In EM, the solution to which the algorithm converges depends on the initial guess. Therefore, in order to have an unbiased estimate, one must choose a point uniformly at random from the space of possible values. Although it may seem reasonable to choose values for each θ_i uniformly in $[0, 1]$ and then normalize them, it turns out that this does not yield uniformly distributed initial guesses. One way to correctly generate the initial guess is to sample $W - 1$ points uniformly from the unit line and then take the difference between adjacent points (including 0 and 1) [20, Chapter XI, Theorem 2.1]. This is equivalent to drawing from the Dirichlet distribution with W parameters $\alpha = (1, \ldots, 1)$, since the Dirichlet PDF at point $\boldsymbol{\theta}$ is proportional to $\prod_{i=1}^{W} \theta_i^{\alpha_i - 1}$.

Nevertheless, such an initialization combined with the other two steps of EM will give us estimates $\hat{\theta}_i \in [0, 1]$ hence producing biased estimates as they are not free to assume any real value. Therefore, it is possible for the EM to achieve an MSE not in agreement with the CRLB we derived previously. This is the case when the number of samples N is small and, consequently, the diagonal of $J^{-1}(\boldsymbol{\theta})$ has relatively large values (possibly

greater than 1). On the other hand, for large N, the number of observed sets with size i will converge to a Normal distribution with mean θ_i and small variance. For small enough variance, restricting θ_i to be between 0 and 1 does not affect the final estimate significantly and thus the CRLB accurately bounds the MSE.

4.7 Related Work

Not much prior work exists in the literature on theoretical bounds for estimation error in problems related to the SSD estimation. Hohn and Veitch [37] first observed that using a sampling probability of $p < 1/2$ poses problems in the context of two specific estimators for the flow size distribution when the distribution obeys a power law. In particular, they showed that their estimators of the flow size distribution are asymptotically unbiased with decreasing error as the number of flow samples increases when $p \geq 1/2$, but not when $p < 1/2$. Our work shows that this is a fundamental result of SSD estimation and not specific to any specific estimator. Ribeiro et al. [75] was the first to introduce the use of Fisher information as a design tool for flow size estimation. Experiments reported in that paper suggested that there is little statistical information about the distribution when p is small and showed how this information can be significantly increased with the addition of other data taken from packet headers. Tune and Veitch [88] applied Fisher information to compare packet sampling with flow sampling. In the process of doing so, they obtained a variety of useful Fisher information inverse identities, some of which we rely on in this work.

In [74], Ribeiro et al. study the problem of predicting the distribution of cascade sizes on a network building on the theoretical results presented in this chapter. Similar to what we did here, the authors show the existence of a big data paradox: on power law networks, as the network size grows, increasing both the available historical data and the maximum cascade sizes, predictions beyond the time horizon of the historical data get more inaccurate, whereas predictions within that time horizon become more accurate.

4.8 Conclusions

In this chapter we give explicit expressions of MSE lower bounds of unbiased estimators of the distribution of set-sizes θ and the average set-size $m_{\boldsymbol{\theta}}$ with sampling probability p. We show that the estimation error of $\boldsymbol{\theta}$ grows at least exponentially in W, when $-\log \theta_W = W \log a + o(W)$ as $W \to \infty$ for some $0 < a < 1$, and $p < a/(a+1)$, or when $-\log \theta_W = o(W)$ as $W \to \infty$ and $p < 1/2$, which indicates that unbiased estimators of some distributions $\boldsymbol{\theta}$ are too inaccurate to be useful for practitioners. Moreover we show that unbiased estimates of m_θ suffer from similar problems.

CHAPTER 5

SELECTIVE HARVESTING OVER NETWORKS

5.1 Introduction

Networked active search [27, 57, 60, 91] is a technique for finding the largest number of *target nodes* – i.e., nodes with a target label – in a network by querying nodes in a weighted graph, under a query budget constraint. Nodes have hidden labels but the network topology and edge weights are **fully observable** and **any node** can be queried at any time. Edge weights encode some form of node similarity that can be used to improve querying efficiency. Unfortunately, edge weights, network topology and node information are rarely available to be downloaded from one centralized place (except for the company that owns the network). As a result, today's prevalent method to collect network data is to query neighbors of already queried nodes (crawling). Like networked active search, other similar techniques, such as learning to crawl [31, 68], also assume that edge weights between the queried nodes and their neighbors are observed. But in a variety of network crawling problems, such as crawling online social networks, (micro) blog networks, and citation networks, a node query often reveals only node attributes. This process poses an entirely new set of challenges for networked active search and other similar methods.

In this chapter we introduce *selective harvesting*, where the goal is the same as in active search, but in addition to the fixed budget our node querying is subject to a partial – and evolving – understanding of the network. More precisely, we assume that knowledge about the network is restricted to the set of queried nodes and their connections to the rest of the network. Selective harvesting starts from a seed node (typically a target) and proceeds by querying nodes from the *fringe set*, i.e. neighbors of already queried nodes. Training

a classifier for *selective harvesting* is a challenging task due to the fact that the classifier must be trained over observations that depend on previous choices of the same classifier, the hidden network topology, and the distribution of node features over the network. We call this the *tunnel vision effect*. Unlike standard active search, *selective harvesting* has no recourse to true randomness or sample independence that can ease the tunnel effect. Under partially observed networks, traditional active search methods perform quite poorly.

We discover that it is possible to collect a much larger set of target nodes by using multiple classifiers, not by combining their predictions as a weighted ensemble, but switching between classifiers used at each step, as a way to ease the tunnel vision effect. We show that switching classifiers collects more target nodes by (a) diversifying the training data and (b) broadening the choices of nodes that can be queried in the future. Based on these observations, we propose Directed-Diversity Dynamic Thompson Sampling (D^3TS), a Multi-Armed Bandit (MAB) algorithm for non-stationary stochastic processes that intelligently selects a classifier at each step to decide which neighbor to query. Unlike typical MAB problems, where there is a clear exploration and exploitation tradeoff, the standard MAB approach, which forces convergence to the "best classifier", would be suboptimal in the presence of the tunnel vision effect. This gives rise to what we refer as *exploration, exploitation, and diversification* tradeoff. D^3TS ensures continual diversification by using multiple distinct classifiers, which plays a similar role to sample independence and eases the tunnel vision effect.

Interestingly, we find that even a round-robin selection of distinct classifiers often performs better than just using the best classifier or the best active search method for each dataset. Consider simulation results shown in Figure 5.1 (the simulation is further explained in Section 5.6.1, for now we focus only on the overall results). Figure 5.1 shows the number of queries (x-axis) against the number of target nodes found in the CiteSeer network (NIPS papers as targets) normalized by the number of target nodes found by a round robin selection of five distinct simple classifiers (y-axis); the details of these simple

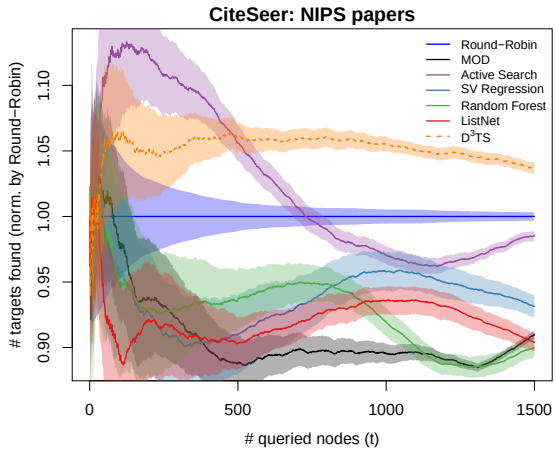


Figure 5.1. Lines show the (scaled) average number of targets found by round-robin, five naïve classifiers and D^{3}TS against the total number of queries (t). Shadows indicate 95% confidence intervals over 80 runs, each starting at a seed uniformly chosen from target population. Surprisingly, round-robin use of five classifiers (including poor-performing ones) outperforms any single classifier in the CiteSeer network. We also see that the best-performing active search method (Wang et al. [91]) has its relative accuracy eroded over time (and we will see why this is likely due to the *tunnel vision effect*). We include the proposed method (D^{3}TS) results, which are consistently better than all competing methods for $t \geq 500$.

classifiers are given in Section 5.3. Note that over time the cumulative gain of the best active search method for this dataset (Wang et al. [91]) slowly erodes until it is worse than the naïve round-robin approach. Our analysis shows that this erosion can be attributed to the tunnel vision effect. Each of the five simple classifiers when used on their own are consistently outperformed by the round-robin approach, and the best such classifiers also suffer from a performance erosion over time. In contrast, our proposed method, D^{3}TS, consistently and significantly outperforms state-of-the-art methods, the round-robin approach, and naïve approaches.

5.1.1 Contributions

The contributions of this chapter are:

1. **Formulation and characterization of Selective Harvesting and Classifier Diversity:** We introduce selective harvesting and show that existing heuristics such as ac-

tive sampling [14, 69] and active search [27, 57, 91] perform poorly in these settings. We show that switching between various classifiers is helpful to achieve greater performance. This works not because we are exploring classifiers in order to find the best one or because we are combining their predictions as an ensemble. Instead, classifier diversity – i.e., the use of multiple classifiers – helps improve accuracy in two complementary ways. It achieves *fringe set diversity*, by exploring regions and thus avoiding remaining in a region where target nodes have been depleted. It also achieves *training sample diversity*, where diverse classifiers create enough diversity of observations to ease the *tunnel vision effect*.

2. **Directed Diversity Dynamic Thompson Sampling (D^{3}TS)**: we propose D^{3}TS, a method for selective harvesting problems which combines different classifiers, and show that it consistently outperforms state-of-the-art methods. We evaluate the proposed framework on several real-world networks and observe that D^{3}TS outperforms all tested methods on five out of seven datasets and exhibits similar performance on the other two.

5.1.2 Outline

The rest of this chapter is structured as follows. In Section 5.2 we formalize the selective harvesting problem and present a generic algorithm for solving it. In Section 5.3 we describe existing and potential approaches to solve this problem and show that the tunnel vision effect hurts their performance. In Section 5.4 we investigate why classifier diversity – i.e., using multiple classifiers – can mitigate the tunnel vision effect. We propose D^{3}TS in Section 5.5. Datasets and results of our evaluation are described in Section 5.6. Related work is described in Section 5.7. We discuss some ideas not explored in this dissertation in Section 5.8. Last, conclusions are presented in Section 5.9.

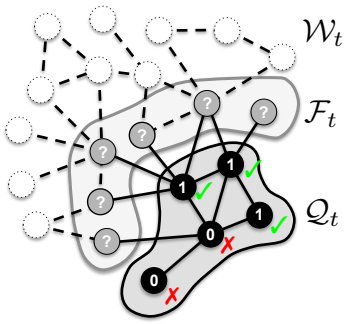


Figure 5.2. Representation of the search state over an unknown graph $\mathcal{G}$ after $t = 5$ steps. Solid nodes and edges show the subgraph $\widetilde{\mathcal{G}}_t$. Black nodes represent queried nodes. Unknown labels of nodes in $\mathcal{F}_t$ are represented by a question mark "?".

5.2 Problem Formulation

In this section we formalize the selective harvesting problem and introduce notation used throughout this chapter. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote an undirected graph representing the network topology. Each node $v \in \mathcal{V}$ has $|\mathcal{L}|$ attributes (domain-related properties of the nodes) encoded without loss of generality as an attribute vector $\mathbf{a}_v \in \mathbb{R}^{|\mathcal{L}|}$.

In networked active search problems, the goal is to find a large set of nodes in $\mathcal{V}$ that satisfy a given search criterion (e.g., nodes that exhibit a given attribute) under the constraint that no more than B nodes can be queried. The search criterion is a boolean function $f : \mathcal{V} \to \{0, 1\}$. Formally, let $\mathcal{V}_+ \subset \mathcal{V}$ be the set of all target nodes, i.e. all v such that $f(v) = 1$. We define node labels y_v as

$$y_v = f(v) = \begin{cases} 1 & \text{if } v \in \mathcal{V}_+, \\ 0 & \text{otherwise.} \end{cases} \qquad \forall v \in \mathcal{V}$$

Selective harvesting is a variant of active search. In active search, the topology is assumed to be known. In selective harvesting, the search is subject to a limited but evolving knowledge of the network. This knowledge is expanded by querying nodes in $\mathcal{V}$, which reveals their labels, neighbors and attribute vectors. A set of pre-queried nodes $\mathcal{Q}_0 \subset \mathcal{V}$ is

given as input (typically consisting of one target node). Subsequent queries are restricted to neighbors of already queried nodes.

At any step t, nodes belong to one of three sets: $\mathcal{Q}_t$, the set of previously queried nodes; $\mathcal{F}_t$, the set of neighbors of queried nodes that have not been queried (referred as fringe nodes or fringe set); or $\mathcal{W}_t$, the set of unobserved nodes, which are invisible to the algorithm. Figure 5.2 illustrates a snapshot of the search process (see caption for details).

Let $\widetilde{\mathcal{G}}_t = (\mathcal{Q}_t, \widetilde{\mathcal{E}}_t)$ denote the subgraph of $\mathcal{G}$ given by the subgraph induced by nodes in $\mathcal{Q}_t \cup \mathcal{F}_t$ minus edges in the subgraph induced by $\mathcal{F}_t$ (i.e., $\widetilde{\mathcal{G}}_t$ contains all edges between nodes in $\mathcal{Q}_t$ plus edges connecting $\mathcal{Q}_t$ to $\mathcal{F}_t$). The graph $\widetilde{\mathcal{G}}_t$ is the portion of the network visible at step t. In $\widetilde{\mathcal{G}}_t$, label y_v is only known for nodes in $\mathcal{Q}_t$.

5.2.1 Generic solution

Given an *initial input graph* $\widetilde{\mathcal{G}}_0$, a selective harvesting algorithm must decide at each step $t = 1, \ldots, B$ what *action* to take, i.e., what fringe node $v \in \mathcal{F}_t$ to query, given the currently available network information. This *action* returns v's label, attributes and connections, which is included as *additional input* to the search in step $t+1$. Node v' label (0 or 1) can be thought of as the *payoff* obtained by querying that node. The algorithm's *output* is the list of target nodes found in B steps. The best algorithm is the one that yields the largest total payoff, i.e., yields the largest number of target nodes.

5.3 Background

In this section, we review methods for searching networks that can be used for or adapted to selective harvesting. These methods exploit correlation between labels of connected nodes to find targets. In addition, we review statistical models that could be used as an alternative (data-driven) approach. In contrast to existing methods, base learners can leverage node attributes by training a statistical model to infer the node's label from the observed graph. As a slight abuse of terminology, we may refer to existing methods and

	PNB [69]	SN-UCB1 [14]	MOD [8]	AS [91]	D^{3}TS (ours)
Unknown network	✓	✓	✓	-	✓
Uses node features	-	-	-	-	✓
Unknown neighbor attributes	-	-	-	-	✓
Fits model to evolving observations	-	✓	-	-	✓
Scalable	-	✓	✓	✓	✓

Table 5.1. Comparison of heuristics for selective harvesting: Active Sampling (PNB), Social Network UCB1 (SN-UCB1), Maximum Observed Degree (MOD), and Active Search (AS).

base learners generically as **classifiers**, since both are used to classify fringe nodes as either targets or not.

5.3.1 Existing methods

There is related work in the literature that provides methods that can be used for or adapted to selective harvesting. A subclass of selective harvesting methods known as active sampling [14,69] does not account for node attributes. Our problem is closely related to the graph-theoretic myopic budgeted online covering problem [8, 16, 43]. In this problem, all nodes are relevant (equivalently, all nodes are targets) and the task is to find a connected set of nodes that yields the largest cover (i.e., the largest set $\mathcal{Q}_t \cup \mathcal{F}_t$). The closest problem to ours is that addressed by networked active search [27, 57, 60, 91], where nodes have hidden labels but the topology and edge weights are **fully observed** and **any node** can be queried at any time. Algorithms for myopic budgeted online covering and active search can be adapted for selective harvesting; active sampling methods require little or no modification.

We adapt four representative methods of the above to selective harvesting, namely Active Sampling [69] (PNB – in reference to the authors surnames), Maximum Observed Degree (MOD) [8], Social Network UCB1 (SN-UCB1) [14], and Active Search (AS) [91]. Table 5.3 summarizes the key differences between these methods and the proposed method D^{3}TS.

5.3.1.1 Active Sampling (PNB)

PNB is a representative algorithm from the class of networked active sampling approaches proposed in [69]. PNB estimates a fringe node's payoff value y_v using a weighted average of the payoffs of observed nodes two hops away from v, where weights are the number of common neighbors with v. Fringe nodes are included among these observed nodes, requiring all payoffs to be collectively estimated by a label propagation-like procedure based on Gibbs Sampling. PNB also tracks a running average of payoff values acquired from random jumps, which we do not allow in our simulations since these are not possible in selective harvesting. We refer the reader to [69] for a complete description of PNB and its parameters.

5.3.1.2 Social Network UCB1 (SN-UCB1)

The SN-UCB1 search algorithm proposed in [14] divides fringe nodes into equivalence classes and samples from theses classes using a multi-armed bandit (MAB) algorithm. Equivalence classes are composed of all fringe nodes connected to the same set of queried nodes. These classes are volatile: they split, disappear and appear over time, requiring the use of a variant of the UCB1 algorithm called VUCB1. Although this method learns about the equivalence classes, it does not learn a statistical model, and does not account for node attributes. Similar to selective harvesting, it assumes partial but evolving knowledge about the network.

5.3.1.3 Maximum Observed Degree (MOD)

MOD is a myopic algorithm proposed in [8] to maximize the network cover as it explores a graph. MOD is the optimal greedy cover algorithm in a finite random power law network (under the Configuration Model [66]) with degree distribution coefficient either one or two. In our simulations we adapt MOD to select the fringe node with the maximum number of target neighbors in the queried set (ties are resolved randomly). From the ex-

pected excess degree results in [8] such fringe nodes are rich with target neighbors provided the underlying network exhibits strong homophily with respect to node labels.

5.3.1.4 Active Search

The Active Search method proposed by Wang et al. [91] attempts to find target nodes by assuming that labels are defined by a smooth function over the graph edges. To estimate the unknown labels, it attaches to each labeled instance a virtual node containing the instance's label and then performs label propagation on the original graph. This method assumes that the underlying graph is known, which allows it to estimate the future impact of choosing a given fringe node. We adapt Active Search to run label propagation only on the observed graph.[1]

5.3.2 Data-driven methods

A data-driven selective harvesting algorithm trains a statistical model to estimate the expected payoff $\mu_t(v)$ obtained from querying fringe node $v \in \mathcal{F}_t$, based on v's relationship with the observed graph $\widetilde{\mathcal{G}}_t$ at step t. We encode this relationship as a "local" feature vector $\mathbf{x}_{v|\widetilde{\mathcal{G}}_t}$, which we describe next. Note that v's features differ from v's attributes (denoted by $\mathbf{a}_v$). Since v's attributes are not observable until it is queried, we compute v's local features from the observed graph $\widetilde{\mathcal{G}}_t$ to use as training data for base learners.

5.3.2.1 Feature Design

We define features for each fringe node in $v \in \mathcal{F}_t$. They are divided into:

- **Pure structural features**: observed degree and number of triangles formed with observed neighbors.

[1] Although the method proposed by Wang et al. [91] is outperformed by a more recent proposal [57] in active search problems, we found the opposite to be true when the graph is not fully observable. In addition to being highly sensitive to the parameterization, the most recent method computes and stores a dense correlation matrix between all visible nodes, which is hard to scale beyond 10^5 nodes.

- **Structure-and-attribute blends**: number and fraction of target neighbors, number and fraction of triangles formed with two non-target (and with two target) neighbors, number and fraction of neighbors mostly surrounded by target nodes, fraction of neighbors that exhibit each node attribute, probability of finding a target exactly after two random walk steps from fringe node.[2]

We build upon features typically used in the literature [78, 79]. We also use a Random Walk (RW) transient distribution to build features: we consider the expected payoff observed by a RW that departs from node $u \in \mathcal{F}_t$ and performs two steps, given by

$$x^{(\text{RW})}_{u|\widetilde{\mathcal{G}}_t} = \frac{\sum_{(u,v)\in\widetilde{\mathcal{E}}_t} \sum_{(v,w)\in\widetilde{\mathcal{E}}_t, w\in\mathcal{Q}_t} y_w}{C_{u|\widetilde{\mathcal{G}}_t}} \tag{5.1}$$

where $C_{u|\widetilde{\mathcal{G}}_t}$ is the number of such paths of length two in $\widetilde{\mathcal{G}}_t$. Note that the RW is not restricted to the immediate neighbors of u. Also, this is not an average among the nodes two hops away from u; this feature depends on the connectedness of the fringe node's neighborhood in the observed graph.

5.3.2.2 Base Learners

The feature vector described above can be given as input to any learning method able to generate a ranking of fringe nodes. We consider classification, regression and ranking methods as suitable candidates for this task. The classification representatives include **Logistic Regression** and **Random Forests**, because they provide ways to rank fringe nodes according to how confident the model is that each fringe node is a target. **Exponentially Weighted Least Squares** (EWLS) and **Support Vector Regression** are included by modeling the task as a regression problem, and the list-wise learning-to-rank method **List-**

[2]Other seemingly obvious features (e.g., number of non-target nodes) are not considered due to colinearity. Longer random walk paths are be too expensive to be used in most real networks.

Net [17] for directly outputting ranks. We briefly describe EWLS and ListNet below and refer the reader to [24] for descriptions of other methods.

5.3.2.2.1 Exponentially Weighted Least Squares (EWLS): computes weights $\mathbf{w}$ that, given a forgetting factor $0 \ll \beta \leq 1$ and regularization parameter λ, minimize the loss function

$$\sum_{i=1}^{t} \beta^{t-i} |y_t - \mathbf{x_t}^\top \mathbf{w}|^2 + \beta^t \lambda \|w\|^2.$$

EWLS gives more weight to recent observations. The weights $\mathbf{w}$ are suitable for fast online updates [56, Section 4.2]. Setting $\beta = 1$ reduces EWLS to ℓ_2-regularized Linear Regression.

5.3.2.2.2 ListNet: this is a representative method from the list-wise approaches for learning to rank (a Machine Learning task where the goal is to learn how to rank objects according to their relevance to a query) [17]. It assumes that the observed ranking $\boldsymbol{\pi}$ is a random variable that depends on the objects' scores (where π_1 is the top-ranked object). The scores are determined by a neural network that is trained by minimizing the K-L divergence between the probability distribution over $\hat{\boldsymbol{\pi}}$ and the probability distribution over a ranking $\boldsymbol{\pi}$ derived from ground-truth scores. In our context, $P(\boldsymbol{\pi})$ is given by

$$P(\boldsymbol{\pi} = \langle \pi_1, ..., \pi_{|\mathcal{F}_t|} \rangle) = \prod_{i=1}^{|\mathcal{F}_t|} \left[\exp(y_{\pi_i}) / \sum_{j=i}^{|\mathcal{F}_t|} \exp(y_{\pi_j}) \right].$$

Since the goal is not to predict the object-wise relevance, all of the statistical power of this method goes into learning the ranking.

As with any learning approach, in the "small data" regime (few observations collected) a base learner may perform worse than heuristic methods that assume homophily w.r.t.

Methods	Datasets (budget B)						
	CS (1500)	**DBP** (700)	**WK** (400)	**DC** (100)	**KS** (700)	**DBL** (1200)	**LJ** (1200)
PNB	**833.2***	260.6*	107.7*	24.3*	178.3*	599.5*	632.4*
SN-UCB1	568.9*	272.3*	71.8*	23.2*	133.2*	399.1*	573.7*
MOD ✓	746.8*	403.0*	140.9*	35.7*	159.6*	580.3*	584.1*
Active Search ✓	808.9*	412.2*	**143.4**	22.6*	215.3*	684.9*	654.2*
Logistic Regression	764.5*	452.5	86.2*	35.8	122.1*	**744.4**	732.0
Random Forest ✓	738.5*	454.0*	127.2*	37.2	215.6*	725.4	728.3*
EWLS	808.2*	**462.4**	82.5*	35.2*	142.3*	656.9*	694.4*
SV Regression ✓	770.6*	456.3*	85.0*	**37.6**	205.3*	**757.1***	736.1
ListNet ✓	742.0*	448.0*	92.5*	34.4*	146.3*	730.7	**742.8**
Round-Robin (all ✓)	822.2*	454.5*	135.3*	37.3	**234.9***	696.0*	716.0*
D^{3}TS (all ✓)	**851.2**	**464.0**	**144.7**	**37.9**	**247.6**	729.5	**737.3**
Target population size	1583	725	202	56	1457	7556	1441

Table 5.2. Average number of targets found by each method after B queries based on 80 runs. **Datasets. CS**: CiteSeer, **DBP**: DBpedia, **WK**: Wikipedia, **DC**: DonorsChoose, **DBL**: DBLP, **KS**: Kickstarter and **LJ**: LiveJournal. Budget B is respectively set to number of targets $\times 1, \times 1, \times 2, \times 2, \times \frac{1}{2}, \times \frac{1}{6}, \times \frac{5}{6}$ truncated to hundreds. First four rows correspond to existing methods; five subsequent rows are base learners. Round-Robin and D^{3}TS combine methods indicated by (✓). Means whose difference to D^{3}TS's is statistically significant at the 95% confidence level are indicated by (*). Best two results on each dataset are shown in bold. **Parameters.** PNB: same as in [69]; Active Search: same as in [91]; ELWS: $\beta = .99$, $\lambda = 1.0$; Logistic Regression and SV Regression: penalty C set using fast heuristic implemented in R package `LiblineaR` [35]; Random Forest: no. variables $= \sqrt{\text{no. features}}$, number of trees $= 500$ for CS, DBP, WK, DC and $= 100$ for KS, DBL, LJ; ListNet: no. iterations $= 100$, tolerance $= 10^{-5}$.

node labels. To mitigate issues related to fitting a learner to few observations and yet allow a fair comparison with the heuristic methods, we query the first 20 nodes using MOD.[3]

To highlight the tunnel vision effect and show how classifier diversity can mitigate it we conduct a large set of simulations. We simulate searches using four heuristics – MOD, PNB, Social Network-UCB1 (SN-UCB1) and Active Search – and five base learners – Logistic Regression, Exponentially Weighted Least Squares (EWLS), Support Vector Regression, Random Forest and ListNet – on seven networks and summarize the results in

[3]In comparison to other combinations of length and heuristic used in the "cold start" phase, this was found to work best.

Table 5.2 (network datasets and target populations are described in Section 5.6.1). We then consider a set of classifiers $\mathcal{M}$ that typically exhibit good performance and cycle between them during the search, in a Round-Robin (RR) fashion.

Based on Table 5.2, we pick $\mathcal{M}$={MOD, Active Search, Support Vector Regression, Random Forest, ListNet}.[4] We use this set of classifiers **throughout the rest of this chapter**, unless otherwise noted. At each step, one of the classifiers in $\mathcal{M}$ is used to determine which node is the most likely to be a target. This node is then queried and the resulting observation is, in turn, used to update all classifiers. One might expect RR's performance to be the average of the performance results yielded by the standalone counterparts, but as we observe in Figure 5.1, this is not the case. Interestingly, switching classifiers at each step outperforms the best classifier in $\mathcal{M}$ on the CiteSeer and Kickstarter datasets, and finds at least 92% as many target nodes as the best classifier on other datasets. In what follows we investigate why the use of multiple classifiers can improve selective harvesting's performance.

5.4 Leveraging diversity through the use of multiple classifiers

We observe that RR outperforms all five classifiers on CiteSeer (Fig. 5.1). Consequently, at least one of them must perform better under RR than on its own. In order to identify which ones do, we show in Figure 5.3 the hit ratio – number of target nodes found divided by number of queries performed using each classifier up to time t – under RR and when used by itself, averaged over 80 runs. Interestingly, after $t = 400$ all classifiers exhibit similar (relative difference smaller than 10%) or better performance under RR than when used alone.

We propose two hypotheses to explain this performance improvement:

[4]We choose MOD in lieu of PNB because MOD is orders of magnitude faster. Among the base learners, we choose one representative of regression (SV Regression), classification (Random Forest) and ranking (ListNet) methods.

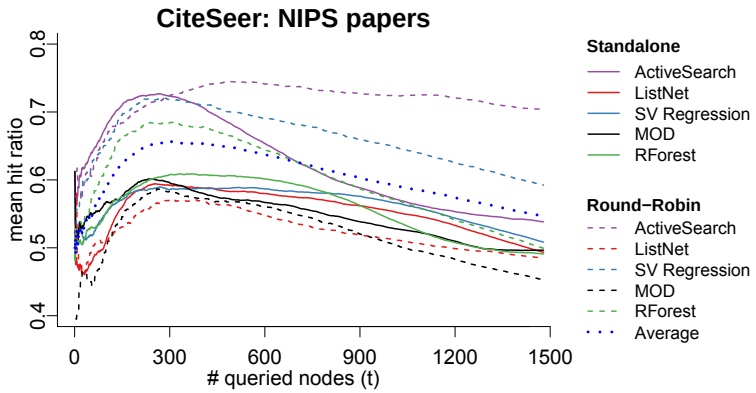


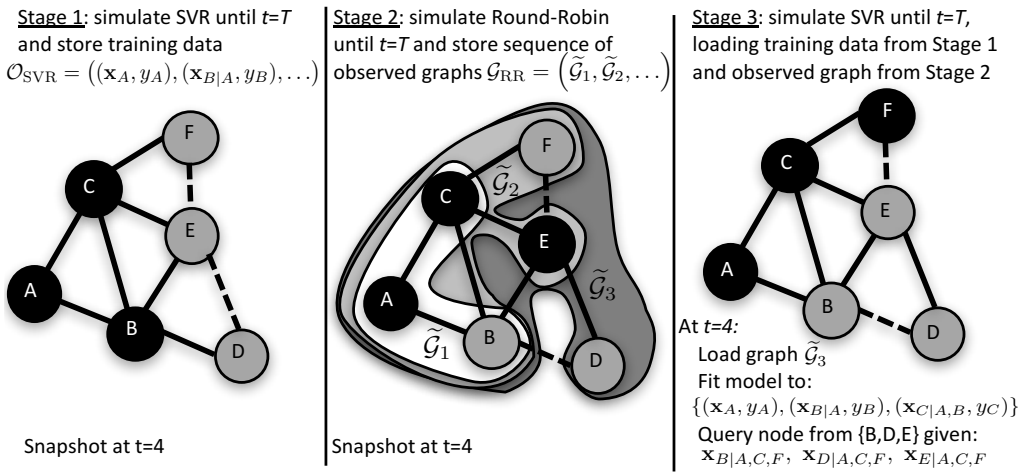
Figure 5.3. Round-robin can have higher hit ratios for each of its classifiers than their standalone counterparts.

(a) **Fringe Hypothesis**: RR explores regions of the graph containing more targets that are likely to be scored high by a classifier, i.e. RR infuses diversity in the fringe set.

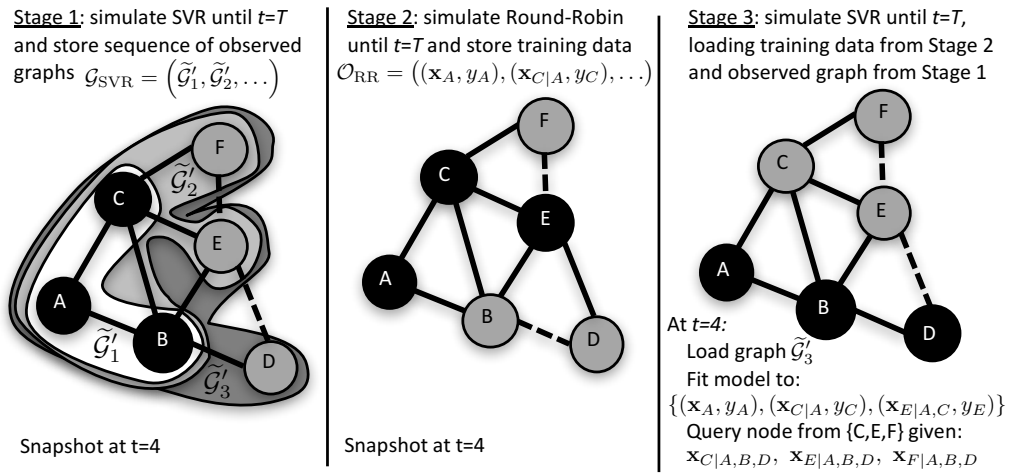
(b) **Training Hypothesis**: Observations from different classifiers can be used to train the others to generalize better and cope with self-reinforcing sampling biases, i.e., diversity in the training set produce a classifier that is better at finding target nodes.

Note that these hypotheses are not mutually exclusive. In what follows, we perform controlled simulations to isolate and study each hypothesis.

Training set diversity directly impacts model parameters. Model parameters, in turn, determine how the fringe set will change. Therefore, to assess the impact of training set diversity we must hold the fringe set diversity constant and vice-versa. This is the key idea behind the two controlled sets of simulations described next. To perform them, we instrumented our simulator to load, from another simulation run, (i) the feature vector $\mathbf{x}_{\sigma_t|\widetilde{\mathcal{G}}_t}$ of node σ_t queried in step t, and label y_{σ_t}, and (ii) the observed graph $\widetilde{\mathcal{G}}_t$ at each step t. In what follows, we show the results obtained using the support vector regression (SVR) model. We denote node σ_t's feature vector and label simply by $\mathbf{x}_t$ and y_t, respectively, to make it easier to follow.



(a) Simulations for studying Fringe Hypothesis



(b) Simulations for studying Training Hypothesis

Figure 5.4. (a) We study the Fringe Hypothesis by recreating the sequence of SVR models from the original simulation run (stage 1) and using them to query nodes on a sequence of observed graphs collected using round-robin (stage 2). (b) We study the Training Hypothesis by recreating the sequence of observed graphs from the original simulation run (stage 1) and using a SVR trained on the samples collected using round-robin (stage 2) to query nodes.

5.4.1 Fringe Hypothesis

Our experiment consists of three stages (Fig. 5.4(a)). First, we store the sequence of observations (i.e., pairs feature vector, label) $\mathcal{O}_{\mathrm{SVR}} = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_B, y_B))$ corresponding to nodes queried when searching a network dataset $\mathcal{D}$ using SVR. Second, we store the sequence of observed graphs $\widetilde{\mathcal{G}}_{\mathrm{RR}} = \left(\widetilde{\mathcal{G}}_1, \ldots, \widetilde{\mathcal{G}}_B\right)$ when searching $\mathcal{D}$ by cycling between models in the set $\mathcal{M}$. Last, we simulate another SVR-based search on $\mathcal{D}$, loading the observed graph at each time step t from $\widetilde{\mathcal{G}}_{\mathrm{RR}}$. However, instead of training the SVR model with observations collected on that run (which most likely differ from those collected during the first stage), we gradually feed it with observations from $\mathcal{O}_{\mathrm{SVR}}$, one for each simulation step t. Therefore, we will reproduce the sequence of classifiers from the first stage, but subject to a different sequence of observed graphs.

5.4.2 Training Hypothesis

As before, our experiment consists of three stages (Fig. 5.4(b)). In the first stage, we store the sequence of observed graphs $\widetilde{\mathcal{G}}_{\mathrm{SVR}} = \left(\widetilde{\mathcal{G}}'_1, \ldots, \widetilde{\mathcal{G}}'_B\right)$ when searching $\mathcal{D}$ using a SVR model. Second, we store the sequence of observations $\mathcal{O}_{\mathrm{RR}} = ((\mathbf{x}'_1, y'_1), \ldots, (\mathbf{x}'_B, y'_B))$ collected when searching $\mathcal{D}$ by cycling among classifiers in $\mathcal{M}$. Last, we simulate another SVR-based search, loading the observed graph at each time step t from $\widetilde{\mathcal{G}}_{\mathrm{SVR}}$, but feeding it observations from $\mathcal{O}_{\mathrm{RR}}$, one by one. Hence, the classifier is fit to a different set of observations, but the search is subject to the same sample path as the SVR-based search from the first stage.

Figure 5.5 contrasts the average number of target nodes found by the original SVR-based search on CiteSeer against those obtained in each set of simulations based on 80 runs. The 95% confidence intervals for the mean at $t = 700$ are $[393.8, 413.1]$, $[416.6, 427.5]$ and $[417.1, 436.7]$. These statistics corroborate the hypotheses that the fringe set and the training data collected by the round-robin policy contribute to improving the performance of the SVR model.

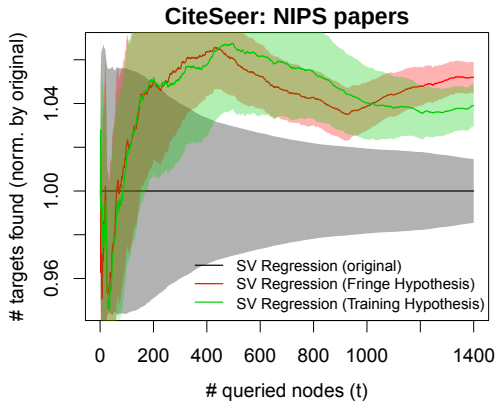


Figure 5.5. SVR classifier and two ways to ease the *tunnel vision effect*: **fringe set diversity** and **training set diversity** improve performance by ensuring greater diversity in query choices and by diversifying the training data, respectively.

Intuitively, when a base learner is fit to the nodes it queried, it tends to specialize in one region of the feature space and the search consequently only explores similar parts of the graph, which can severely undermine its potential to find target nodes. One way to mitigate this overspecialization would be to sample nodes from the fringe set probabilistically, as opposed to deterministically querying the node with the highest score. This alternative is investigated in Appendix H, where the ranking associated with each classifier is mapped into a probability distribution. The results show no significant performance improvement over those obtained when a single classifier chooses nodes to query deterministically. It is possible that using node scores attributed by each classifier – rather than node rankings – to obtain a mapping to a probability distribution yields better results, but this is left as future work.

The round-robin policy infuses diversity in the training set without sacrificing performance. This diversity is achieved by "asking another classifier" what is the best node to query at a given step. In scenarios where all classifiers would have performed reasonably well if used alone, learning from another's classifier query is likely to improve one classifier's ability to find targets, especially when they disagree.

Yet, different classifiers inherently exhibit different performances on a dataset. In the following section, we propose a method that learns these inherent performances online and thus improves upon round-robin by not using all classifiers an equal number of times. We call it Directed Diversity Dynamic Thompson Sampling because it is based on the Dynamic Thompson Sampling algorithm for multi-armed bandit problems and because it leverages diversity in a "directed way" as opposed to randomly sampling nodes.

5.5 Directed Diversity Dynamic Thompson Sampling (D^3TS)

Selective harvesting with multiple classifiers can be cast as a Multi-Armed Bandit (MAB) problem. In selective harvesting, the sequential decision problem consists of choosing the node to query at each step, given recommendations from several models. There are two ways of mapping selective harvesting to a MAB problem. The first (and simplest) mapping is context-free. Each model is represented by an arm (i.e., the problem reduces to one of choosing a model at each time step). Models are treated as black boxes that "internally" query a node and return the node's label. The queried node's label is seen as the model's payoff. The second mapping falls into the class of contextual bandits. Each fringe node represents an action and each model represents an expert that provides recommendations on how to choose the actions. Node features correspond to action contexts, which are used by the experts to compute their recommendations.

Despite the potential advantage of accounting for node features directly and combining the advice of several models, most algorithms for contextual bandits assume fixed and small (relative to the time horizon) sets of actions, whereas the fringe set is dynamic and potentially orders of magnitude larger than the query budget. Among context-free bandits, we claim that algorithms for stochastic bandits with non-stationary distributions are the best candidates for combining classifiers in selective harvesting, as we observe that the average hit ratio can drift over time (Fig. 5.3). While adversarial bandits allow payoff distributions to change arbitrarily, they cannot exploit the fact that the mean payoff evolves in

Algorithm 3 D^{3}TS (budget B, model set $\mathcal{M}$, threshold $C \geq 2$)

▷ Assume $\mathcal{B}_t$ is updated after each iteration.
for t in $1, \ldots, B$ **do**
    for k in $1, \ldots, |\mathcal{M}|$ **do**
        $\hat{r}_t^{(k)} \sim \text{Beta}(\alpha_k, \beta_k)$
    $I_t = \arg\max_{k \in 1,\ldots,K} \hat{r}_t^{(k)}$
    $\hat{\mathbf{y}} =$ estimate payoffs using classifier I_t and $\widetilde{\mathcal{G}}_t$
    $b = \arg\max_{v \in \mathcal{B}_t} \hat{y}_v$
    $r_t = y_b = \text{query}(b)$
    if $\alpha_{I_t} + \beta_{I_t} < C$ **then**
        $\alpha_{I_t} = \alpha_{I_t} + r_t$
        $\beta_{I_t} = \beta_{I_t} + (1 - r_t)$
    else
        $\alpha_{I_t} = (\alpha_{I_t} + r_t) \times C/(C+1)$
        $\beta_{I_t} = (\beta_{I_t} + (1 - r_t)) \times C/(C+1)$
    $\mathcal{M} =$ update or retrain classifiers given new point $(x_{b|\widetilde{\mathcal{G}}_t}, y_b)$

a well-behaved manner. A thorough comparison of several bandit algorithms described in Appendix I supports our claim. Our comparison includes the Exp4 and Exp4.P algorithms for contextual bandits, which combine the prediction of all classifiers in a similar way that traditional ensemble methods do.

For the reasons above, we adapt the Dynamic Thompson Sampling (DTS) algorithm [32] proposed for MABs with non-stationary distributions to the selective harvesting problem. DTS is based on the Thompson Sampling (TS) algorithm for stochastic MABs, where each arm $k = 1, \ldots, K$ is modeled as a distribution Beta(α_k, β_k). At step t, TS samples $\hat{r}_t^{(k)} \sim$ Beta(α_k, β_k) and selects the arm with the largest sample, i.e., $I_t = \arg\max_{k \in 1,\ldots,K} \hat{r}_t^{(k)}$. Given the binary payoff r_t received after selecting arm I_t, the distribution parameters are updated according to the Bayesian rule, i.e., $\alpha_{I_t} = \alpha_{I_t} + r_t$ and $\beta_{I_t} = \beta_k + (1 - r_t)$. In essence, DTS normalizes arm k's parameters such that $\alpha_k + \beta_k \leq C$, where C is a bounding parameter. When C is small (we set $C = 5$), DTS prevents Beta distributions from becoming too concentrated around the mean, which will in turn guarantee that we continue leveraging diversity. This highlights an *exploration, exploitation and diversification* tradeoff in selective harvesting that goes beyond the duality found in classic

Dataset	nodes	edges	node attributes	target nodes
DBpedia	places	hyperlinks	place type	admin. regions
CiteSeer	papers	citations	venues	top venue
Wikipedia	wikipages	links	topics	OOP pages
Kickstarter	donors	co-donors	backed projects	DFA donors
DonorsChoose	donors	co-donors	awarded projects	P donors
LiveJournal	users	friendship	enrolled groups	top group
DBLP	authors	co-authorship	conference	top conference

Table 5.3. High-level description of each network.

MAB problems, as simply converging to one arm would be suboptimal. The pseudo-code for our adapted method, D^{3}TS, is shown in Algorithm 3. In what follows we compare D^{3}TS against all approaches for selective harvesting discussed in Section 5.3.

5.6 Simulations

This section describes the datasets used in our simulations, together with simulation results and comparisons with baseline methods.

5.6.1 Datasets

To evaluate the above search methods, we use seven datasets corresponding to undirected and unweighted networks containing node attributes. In the following we describe each of the datasets summarized in Table 5.3. Basic statistics for each network are shown in Table 5.4.

The first three datasets have been used as benchmarks for Active Search [57, 91]. Despite the fact that Active Search assumes that the network topology is known, we can use these datasets to evaluate active search methods by only revealing parts of the graph as the search proceeds.

Dataset	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{L}\rvert$	$\lvert\mathcal{V}_+\rvert/\lvert\mathcal{V}\rvert$
DBpedia	5.00K	26.6K	5	14.5%
CiteSeer	14.1K	42.0K	10	13.1%
Wikipedia	5.27K	64.6K	93	3.83%
Kickstarter	27.8K	2.77M	180	5.27%
DonorsChoose	1.15K	6.60K	284	4.96%
LiveJournal	4.00M	34.7M	5K	0.04%
DBLP	317K	1.05M	5K	2.38%

Table 5.4. Basic statistics of each network: $|\mathcal{V}|$ (number of nodes), $|\mathcal{E}|$ (number of edges), $|\mathcal{L}|$ (number of attributes) and $|\mathcal{V}_+|/|\mathcal{V}|$ (fraction of target nodes).

5.6.1.1 DBpedia

A network of 5000 populated places from the DBpedia ontology formed by linking pairs whose corresponding Wikipedia pages link to each other, in either direction. Places are marked as "administrative regions", "countries", "cities", "towns" or "villages".

5.6.1.2 CiteSeer

A paper citation network composed of the top 10 venues in Computer Science. Papers are annotated with publication venue.

5.6.1.3 Wikipedia

A web-graph of wikipages related to programming languages. Pages are annotated with topics obtained by thresholding a pre-computed topic vector [91].

Two network datasets from the SNAP repository [53] typically used to validate community detection algorithms are also used. We label nodes belonging to the largest ground-truth community as targets. Other community memberships are used to define a binary attribute vector $\mathbf{a}_v \in \{0,1\}^{|\mathcal{L}|}$ for all $v \in \mathcal{V}$.

5.6.1.4 LiveJournal

A blog community with OSN features, e.g.: users declare friendships and create groups that others can join. Users are annotated with the groups they joined.

5.6.1.5 DBLP

A scientific collaboration network where two authors are connected if they have published together. Authors are annotated with their respective publication venues.

Last, we use datasets containing donations to projects posted on two online crowdfunding websites. To assess the performance of each classifier in low correlation settings, we build a social network connecting potential donors where edges are weak predictors of whether or not neighbors of a donor will also donate. We label nodes as targets if they donated to a specific campaign. Historical donation data prior to that is used to build the network and define node attributes.

5.6.1.6 Kickstarter(.com)

An online crowdfunding website. This dataset was collected by GitHub user *neight-allen* and consists of 3.04M donors that together made 5.87M donations to 87.3K projects. We create a donor-to-donor network by connecting donors that donated to the same projects in the past. More precisely, we assume that backers of small unsuccessful campaigns (between 100 and 600 backers) are all connected in a co-donation network – say, their names are published on the campaign's website. We choose campaigns with few donors so that the resulting network is sparse and the network discovery problem challenges D^3TS. Our dataset has 180 small unsuccessful projects between 04/21/2009 and 05/06/2013, containing a total of 27.8K donors. We then choose the 2012 project (denoted DFA) that has the largest number of donors in our dataset. The goal of the recruiting algorithm is to recruit the 2012 DFA donors through the donor-to-donor network of past donations (2009–2011).

5.6.1.7 DonorsChoose(.org)

An online crowdfunding website where teachers of US public schools post classroom projects requesting donations (e.g., for a science project). The dataset is part of the KDD 2014 Cup containing 1.29M donors that together made 3.10M donations to 664K projects from 57K schools. Donations include information such as donor location, donation amount,

Dataset	avg top 5		avg top 3		avg top 1	
	RR	D^{3}TS	RR	D^{3}TS	RR	D^{3}TS
CiteSeer	1.03	1.07	1.01	1.04	0.99	1.02
DBpedia	1.00	1.02	0.99	1.01	0.98	1.00
Wikipedia	1.11	1.19	0.99	1.05	0.94	1.01
DonorsChoose	1.03	1.04	1.01	1.03	0.99	1.01
Kickstarter	1.20	1.27	1.11	1.17	1.09	1.15
DBLP	0.96	1.00	0.94	0.98	0.92	0.96
LiveJournal	0.99	1.01	0.97	1.00	0.96	0.99

Table 5.5. Performance ratios: between RR (D^{3}TS) and average of top $k = 1, 3, 5$ standalone classifiers.

awarded project, among other node features. As donors tend to be loyal to the same schools, we focus on the school that received the most donations in the dataset. We use projects from 2007 to 2012 to construct a donor-to-donor network where an edge exists between two donors if they donated to the same project less than 48 hours apart. We then select the project P in 2013 with the largest number of donations.

5.6.2 Results

In this section, we compare the performances of D^{3}TS, Round-Robin (RR) and standalone classifiers, w.r.t. the number of targets found at several points in time. We set the threshold $C = 5$ in D^{3}TS and parameters of all classifiers as in Table 5.2.

We simulate selective harvesting on each dataset for a large budget B, chosen in proportion to the target population size (e.g., for DonorsChoose we set $B = 300$, for Kickstarter we set $B = 1500$). In order to contrast RR's and D^{3}TS' performance against that obtained if side information about the identity of the top k performing classifiers on a given dataset were available, Table 5.5 lists ratios between RR's (and D^{3}TS') performance and the average performance of the top $k = 1, 3, 5$ standalone classifiers. Note that we consider the top

k from all nine standalone classifiers described in Section 5.3, not only the classifiers used by RR (and D^{3}TS). Top classifiers vary across datasets.[5]

Overall, we observe that RR's performance is comparable to that of the top 3 classifiers and can sometimes outperform them (by up to 11%). In the worst case, RR's performance is 92% of that of the best standalone classifier (DBLP). D^{3}TS consistently improves upon RR and yields results at least as good as the best standalone classifier on all datasets except DBLP and LiveJournal, where its performance is respectively 96% and 99% of that of the best classifier. D^{3}TS outperforms the best classifier by up to 15% (Kickstarter).

We now describe the results for each dataset in detail, except for CiteSeer, which was discussed in the introduction. Figure 5.6 contrasts the average number of targets found by RR and D^{3}TS against those found by standalone classifiers, scaled by RR's performance. We include results for five out of nine classifiers (the same ones used in $\mathcal{M}$) to avoid clutter.

On DBpedia, LiveJournal, DonorsChoose and Kickstarter, even RR was able to outperform the existing methods, except for the initial steps (where absolute differences are small anyway). Moreover, on the first two datasets, base learners outperformed existing methods. However, as shown in DonorsChoose and Kickstarter plots, a data-driven classifier by itself does not guarantee good performance.

On most datasets D^{3}TS matches or exceeds the performance of the best standalone classifier. In particular, on Kickstarter, both RR and D^{3}TS find significantly more target nodes than standalone classifiers. While RR can leverage diversity from using multiple classifiers to avoid the tunnel vision effect, D^{3}TS goes beyond and intelligently decides which classifier to use without harming diversity. To illustrate this, we look at the fraction of times D^{3}TS used a given classifier at turn t in 80 runs. Figure 5.7 shows this time series for DBpedia. From the small fraction of uses, we find that MOD performs poorly not only

[5]We conducted some preliminary studies on dataset characteristics that favor the performance of some classifiers over others. A positive correlation between homphily (measured by the assortative coefficient [65]) and Active Search's performance was observed, but a more detailed investigation is left as future work.

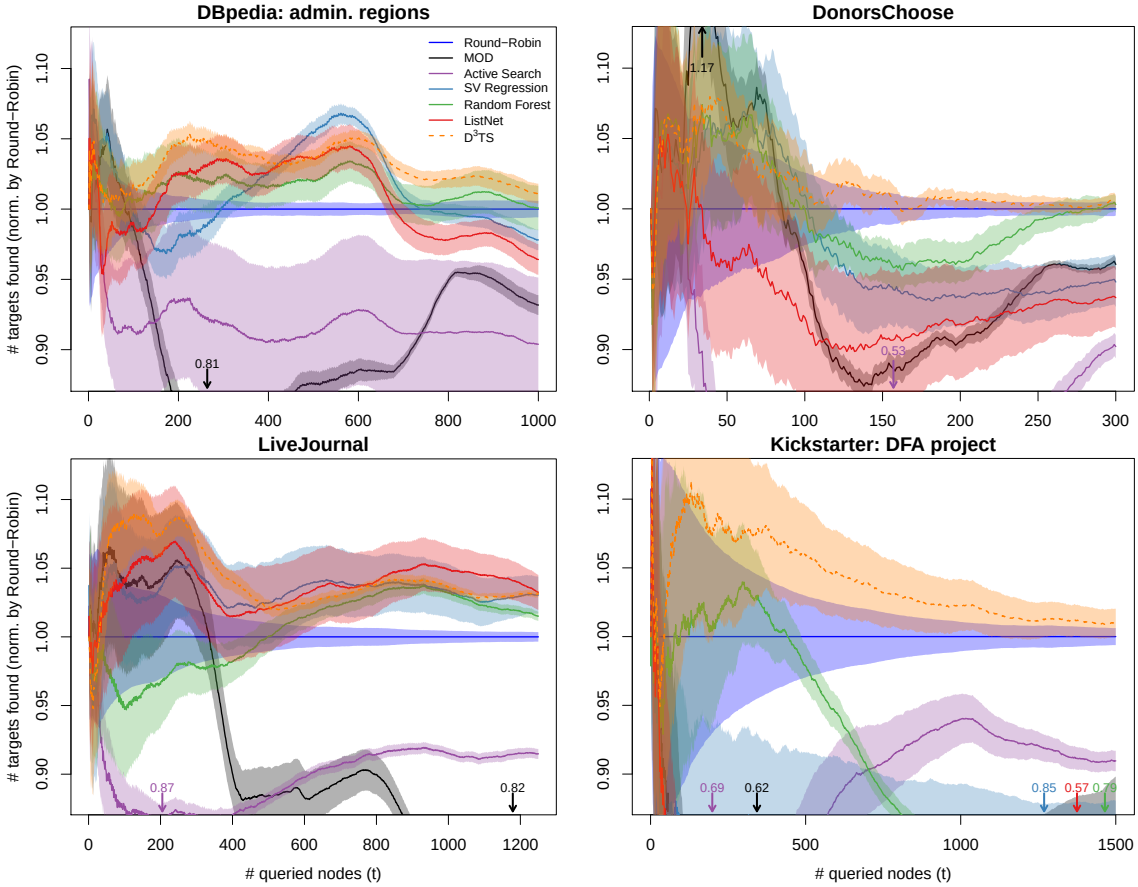


Figure 5.6. Average number of targets found by Round-Robin (RR), D^{3}TS and five standalone classifiers over 80 runs. Shaded areas represent 95% confidence intervals. Arrows indicate minimum values for corresponding colors' classifiers, when off-the-chart. Standalone classifiers are often outperformed by RR. D^{3}TS improves upon RR.

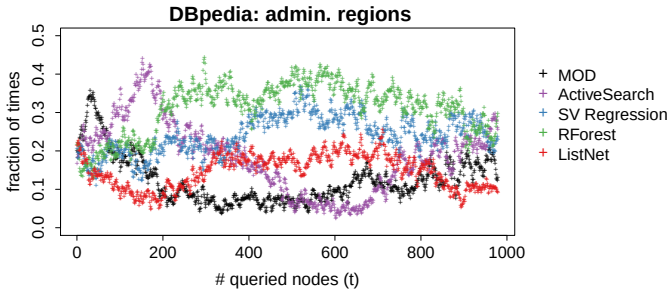


Figure 5.7. D^{3}TS: fraction of runs in which each classifier was used in step t (smoothed over five steps).

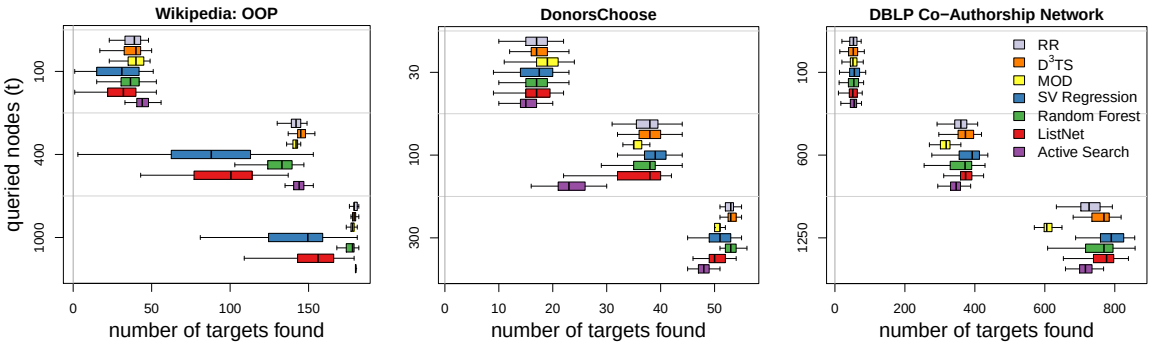


Figure 5.8. RR and D^{3}TS can perform well even when including classifiers that perform poorly as standalone.

on its own, but also when used under D^{3}TS. Fortunately, D^{3}TS can learn classifiers' relative performances and adjust accordingly.

A closer look at the distribution of the number of targets found by each method highlights an important advantage of leveraging diversity. Figure 5.8 shows boxplots of RR and D^{3}TS' performance in each dataset, for several points in time.[6] On Wikipedia, DonorsChoose and Kickstarter, although some of the classifiers used by RR and D^{3}TS yield poor results on their own, RR and D^{3}TS's still attain large mean and low variance. D^{3}TS was only outperformed by a standalone classifier on DBLP (statistically significant). Because DBLP has the largest number of target nodes in the fringe set (on average) over all datasets, classifiers are less likely to be penalized by the tunnel vision effect on DBLP.

5.6.3 Classifier combinations

We also conducted an exhaustive set of simulations where we consider all 31 combinations of these five classifiers under D^{3}TS. We restrict this analysis to a set of networks $\mathcal{D}$ composed of the five smaller datasets. Suppose we had an oracle that could tell which combination of classifiers performs best on a dataset $D \in \mathcal{D}$. We can then define the

[6]The box extremes in our boxplots indicate lower and upper quartiles of a given empirical distribution; its median in marked in between them. Whiskers indicate minimum and maximum values.

(normalized) regret of a classifier set $\mathcal{M}$ on D as

$$R(\mathcal{M}, D) = 1 - \frac{N_+(\mathcal{M}, D)}{\max_{\mathcal{M}'} N_+(\mathcal{M}', D)}$$

where $N_+(\mathcal{M}, D)$ is the number of target nodes found by $\mathcal{M}$ on D. If we define the optimal combination $\mathcal{M}^\star$ to be the one that minimizes the maximum regret, i.e.,

$$\mathcal{M}^\star = \arg\min_{\mathcal{M}} \max_{D \in \mathcal{D}} R(\mathcal{M}, D),$$

then $\mathcal{M}^\star$ indeed includes all five classifiers (maximum regret is 2.8%). Otherwise, if we define the optimal combination $\mathcal{M}^\dagger$ to be the one that minimizes the average regret, i.e.,

$$\mathcal{M}^\dagger = \arg\min_{\mathcal{M}} \sum_{D \in \mathcal{D}} R(\mathcal{M}, D)/|\mathcal{D}|,$$

then $\mathcal{M}^\dagger$ is the combination composed of MOD, Active Search, SVR and Random Forest (average regret is 0.9%). We note, however, that the performance obtained by combination $\mathcal{M}^\star$ on each dataset is at most 0.7% smaller than that obtained by $\mathcal{M}^\dagger$ (CiteSeer). Moreover, we observed that adding a second classifier to a standalone classifier for selective harvesting improves results in about 84% of the cases. This attests to the robustness of using D^{3}TS as the classifier selection policy.

5.6.4 Running time

We measured the average wall-clock time of 80 single-threaded runs of each classifier on an Intel Xeon E5-2660@2.60GHz processor, for five datasets. We do not include measurements for DBLP and LiveJournal because they were simulated in a more heterogeneous environment. In what follows we list inside parentheses the average wall-clock time to find a target (in sec.), for CiteSeer, DBpedia, Wikipedia, DonorsChoose and Kickstarter (in this respective order).

Among standalone classifiers in $\mathcal{M}$, MOD $(0.06, 0.08, 0.14, 0.29, 3.55)$ and Active Search $(0.05, 0.11, 0.17, 0.37, 1.71)$ were the fastest, followed by ListNet $(0.35, 0.31, 1.76, 2.13, 8.42)$, SVR $(0.37, 0.80, 1.26, 5.88, 9.35)$ and Random Forest $(2.54, 4.27, 6.75, 16.75, 43.80)$. We emphasize, however, that MOD and Active Search require no fitting, which is the most expensive step for a base learner. In spite of its good performance at finding target nodes, Random Forest takes much longer than other classifiers to fit and thus, exhibits the longest average time between successful queries.

One of the advantages of D^{3}TS is that it can benefit from Random Forest (and more sophisticated classifiers in general) while only incurring the computational cost for the steps in which they are used. D^{3}TS $(0.54, 1.29, 1.87, 4.98, 25.67)$ exhibits slightly smaller ratios than Round-Robin $(0.64, 1.11, 2.24, 5.21, 16.35)$, except on DBpedia and Kickstarter, where D^{3}TS tends to use Random Forest more often than Round-Robin does. Note that the D^{3}TS running time is determined by the classifiers it uses and their implementations. Replacing methods used in this paper by online counterparts can lead to significant reductions in running time. In particular, Random Forest – which has the largest running time – can, in principle, be replaced by online random forests when bounds on feature values are known in advance.[7]

5.6.5 Dealing with Disconnected Seeds

In the previous simulations, the search starts from a single seed (starting node). When more than one seed is available, the search process may end up exploring various regions of the graph at the same time. In this scenario, the question arises as to how to adequately model the observations in these regions. Intuitively, each region may exhibit distinct characteristics such as target distribution and node degree. Furthermore, some regions may be more similar than others. In some cases, it may be better to fit classifiers to specific re-

[7]We attempted to replace Random Forests by Mondrian Forests [48], but the only publicly available implementation is not optimized enough to be used in our application.

gions of the network where they operate (i.e., using observations collected only from that region), while fitting all classifiers to all observations would probably be the best course if all regions are very similar to each other. One can also consider hierarchical models, which model each region separately but allow some information sharing.

In this section, we consider standalone classifiers and compare their performance in two extreme scenarios: using a single classifier and starting from k seeds (thus modeling all k regions together), or using k models, each initially associated with a single seed (each simulation run uses the same k seeds in both scenarios to reduce variance).

In the multiple classifier scenario, the classifier associated with each region is used to rank its corresponding fringe set at each step t. A single node to be queried must then be selected among all fringe nodes. In particular, we use the EWLS regression model. We select the node with the highest estimated payoff across all rankings, and the model responsible for this estimation is then updated with the new observation.

We vary k from 2 to 6 and observe that, for datasets with a small number of attributes, some improvement is obtained when using multiple classifiers, each with its own model. For instance, on DBpedia, which has only five attributes, an average increase from 523.9 to 562.5 is seen for $k = 3$. However, as the number of node attributes increases, either no significant differences between the average payoffs is observed (Donors, CiteSeer) or the single classifier approach yields better performance (Wikipedia). All comparisons are based on a 95% confidence interval of the mean total payoffs. When D^{3}TS is used in place of standalone classifiers, base learners must be fit to region-specific observations in the case of datasets with few attributes, and fit to the entire training set in the case of datasets with many attributes.

5.7 Related work

The closest work to ours is on networked active search. The goal of active search is to uncover as many nodes of a target class as possible in a network where the topology

is known [27, 28, 57, 60, 91]. Like selective harvesting, active search considers situations where only members of a target class (e.g., malicious class) are sought. Since obtaining labels is associated with a cost (time or money), it is paramount to avoid spending resources on nodes that are unlikely to be targets. Unlike our problem, active search assumes the network topology is known and that any node can be queried at any time.

In [70] a problem similar to selective harvesting is investigated and a learning-based method called Active Exploration (AE) is proposed. Unlike selective harvesting, fringe nodes attributes are assumed to be observable. Since node attributes often carry considerable information about the node's label, AE is not directly comparable with other selective harvesting methods. Our solution differs from AE in that it leverages heuristics in addition to base learners and is applicable to a wider range of applications.

Similarly to selective harvesting, active learning is an interactive framework for deciding what data points to collect in order to train a classifier or a regression model. Unlike active search, (i) its main objective is to improve the generalization performance of a model with as few label queries as possible, and (ii) the set of unlabeled points does not grow based on the collected points. A slew of active learning techniques have been proposed for non-relational data settings, including some tailored for logistic regression [82], for dealing with streamed data [6] and for the case of extreme class imbalance [5]. Although the retrieval of target nodes can benefit from an accurate model, it is unlikely that active learning heuristics (e.g., uncertainty sampling [83]) for training a single classifier can be used for selective harvesting without sacrificing performance. However, it may be possible to adapt active learning techniques proposed for training classifier ensembles (e.g., query by committee [84]) in such a way that, at the same time we collect points on which many classifiers disagree, we ensure that promising candidates among fringe nodes are queried before the sampling budget is exhausted.

Despite these differences, there is an interesting parallel between selective harvesting with many models and a body of research on active learning with a set of active learners (or

heuristics). Both problems can be cast as MABs, where fringe nodes are analogous to unlabeled data points. In active learning, a reward is indirectly related to the collected point: it is computed as some proxy for or estimate of the model's performance on a test set, when fit to all points collected up to a given step. In contrast, rewards in active search are simply the node labels. Like selective harvesting, active learning can either map heuristics directly as arms [11] or map heuristics as experts that give recommendations on how to choose the unlabeled points [38]. In both works it has been observed that combining heuristics may often outperform the single best heuristic. While these works apply algorithms for adversarial bandits to active learning, we find that Dynamic Thompson Sampling for stochastic bandits with non-stationary rewards seem to exploit better the fact that arms rewards are slowly changing in selective harvesting.

Last, another variant of active learning considers the task of learning an ensemble of models [4] or finding a low risk hypothesis $h \in \mathcal{H}$ [25, 26] while labeling as few points as possible. Since the labeled points are biased by the collection process, estimating the models' generalization performances requires either building an uniformly random validation set, or sampling probabilistically at every step and then using importance weighted estimates. In active search, however, the models relative performances can be directly measured from the queried nodes payoffs. Moreover, building a random validation set is bound to degrade performance in scenarios where target nodes are scarce.

5.8 Discussion

In this section, we discuss a few ideas that could not be put into practice or that failed to yield performance improvements.

5.8.1 Accounting for the future impact of querying a node

The active search algorithm assigns a score to each potential query node v that consists of a sum of two terms [91, eq. (2)]: the expected value of v's label and sum of the expected

changes in the labels of all other nodes multiplied by a discount factor $\alpha \ll 1$. The discounted term tries to account for the impact of querying node v, going one step beyond the greedy solution. In selective harvesting, however, our view of the graph is limited to the set of queried nodes and their neighbors, i.e. we cannot compute the impact of choosing a node beyond the fringe set. Even if we could observe the entire graph, accounting for the future impact of querying a node would require us to fit one statistical learning model to each fringe node and predict all the remaining labels at each step, which is too expensive even for a single online model.

5.8.2 Temporal dependencies between observations

We conducted some preliminary experiments that show that EWLS often outperforms ℓ_2-regularized Linear Regression with forgetting factors $\beta > 0.9$ when both use the same regularizing parameter $\lambda \in \{0.1, 1, 10\}$. We propose two non-mutually exclusive hypothesis to explain this phenomenon: (i) the recruitment algorithm induces a temporal dependence that is better represented by giving more weight to recent observations; (ii) the algorithm tends to explore parts of the graph "close" to recently recruited nodes, which represent fringe nodes better than their less recent counterparts due to similarities between nodes that are close in the network. Exploiting this spatial dependence is computationally expensive, as it requires branching several models in a similar fashion to the solution delineated in Section 5.8.1 above. Although EWLS can exploit temporal dependencies induced by recruitment, it is not clear how to perform cross-validation due to the dynamic nature of selective harvesting. In other words, how to test the optimal forgetting factor β if the test set is constantly changing? Also, the value of β that yielded the best results varied across datasets. In the absence of a principled way to perform cross-validation, our recommendation is to combine through D^{3}TS one or more EWLS models – choosing parameters from $\beta \in (0.9, 1.0)$ and $\lambda \in \{0.1, 1, 10\}$ – with other types of models.

5.8.3 Model ensembles

While D^{3}TS makes use of multiple statistical models, only one of them is used for prediction at each step. This differs from model ensembles, which combine predictions of multiple models, possibly with weights. Ensemble methods, such as AdaBoost, are known to perform very well in many classification problems. However, we find that D^{3}TS consistently outperforms AdaBoost. We conjecture that AdaBoost is only slightly less susceptible to the tunnel vision effect than standalone models, as optimizing the weights given to models in the ensemble will eventually nullify the impact of some of them.

5.8.4 Contrasting classifier diversity and diversity in ensembles

Diversity is known to be a desirable characteristic in classifier ensembles [45, 87, 93]. The intuition is that if one can combine accurate models that make independent mistakes, the overall accuracy will be higher than those of the individual models. There are two main classes of techniques for generating diverse ensembles [85]: (i) *overproduce and select*, where a large set of base learners is generated, among which a subset is selected to maximize a given measure of diversity, (ii) *building ensembles*, where the diversity measure is directly used to drive the ensemble creation. In contrast, we did not measure diversity explicitly to select a subset of models or to guide the model generation. This is because the relationship between diversity and overall performance in selective harvesting is more involved. The goal of using diverse classifiers in D^{3}TS is to mitigate the tunnel vision effect. Although each model is fit to the entire training set, diversity is enforced by the use of different types of statistical learning models.

5.9 Conclusions

This chapter introduced selective harvesting, a problem where the goal is to find the largest number of target nodes given a fixed budget and subject to a partial – but evolving – understanding of the network. We discussed existing methods that can be adapted to

selective harvesting and an alternative approach based on statistical models. However, we showed that the tunnel vision effect incurred by the nature of the selective harvesting task severely impacts the performance of a classifier trained on these conditions. We show that using multiple classifiers is helpful in mitigating the tunnel vision effect. In particular, simulation results showed that methods used in isolation often perform worse than when combined through a round-robin scheme. We raised two hypothesis to explain this observation, which were investigated to show that classifier diversity – i.e., switching among classifiers at each querying step – is an important ingredient to collecting a larger set of target nodes in selective harvesting. Classifier diversity increases the diversity of the training set while broadening the choices of nodes that can be queried in the future. Based on these observations we proposed D^3TS, a method based on multi-armed bandits and classifier diversity, able to account for what we named the exploration, exploitation and diversification tradeoff. D^3TS outperforms all competing methods on five out of seven real network datasets and exhibited comparable performance on the other two. While we evaluated D^3TS's performance when used with five specific classifiers (MOD, Active Search, Support Vector Regression, Random Forest and ListNet), the proposed method is flexible and can be used with any set of classifiers (not shown here, replacing SVR by Logistic Regression yields similar results). Moreover, we showed that adding a classifier to a standalone classifier improves selective harvesting results in 84% of the studied cases.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

In this dissertation, we investigated the role of sampling and estimation in different applications related to data science in networks. In the problems investigated in Chapters 3 and 4, sampling is used to estimate characteristics of the population as a whole. In Chapter 5, sampling is used to find nodes in a network that satisfy a given query and, at the same time, to obtain training data to fit statistical models. Unlike classical applications of sampling and estimation, the probability of sampling a node in selective harvesting tasks cannot be computed. Fortunately, there is no need to remove (unknown) biases.

Another difference in these studies lies in the adopted approach or perspective. In Chapters 3 and 5, we propose the DUFS and the D^3TS methods respectively. These methods are designed to account for several practical issues. Although we provide no theoretical guarantees for D^3TS and only some analytical results for DUFS, we conduct a thorough evaluation of these methods through simulation. In particular, our empirical study of selective harvesting sheds light into the tunnel vision effect and how to mitigate it. This allowed us to propose an algorithm that currently matches or exceeds the performance of all competing methods. In contrast to these empirical studies, assuming a simple independent edge sampling model in Chapter 4, allowed us to establish strong theoretical results that hold in the limit for any unbiased estimator of the set size distribution (and related estimation problems).

The approaches that we took to solve the research problems associated with each application illustrate different stages in the typical life cycle of a data science problem. Initially, a task is defined, as well as the way in which data will be collected and presented, heuristics

are proposed for solving the task, hypotheses are posed and a large volume of experiments conducted to validate or falsify the hypotheses, a solution is carefully tailored based on the accepted hypotheses and then evaluated through more experiments. This stage corresponds to our current understanding of selective harvesting over networks. Later, some analytical results and theoretical guarantees are proven for proposed methods, such as the ones we proved for DUFS. As researchers reach a better understanding of the task in hand, it becomes crucial to consider the problem from a more theoretical perspective in order to understand its fundamental limits and – hopefully – how to reach the optimal performance. This stage is exemplified by our work on the set size distribution estimation.

Therefore, general future directions in this line or research include proposing new heuristics, proving properties of existing methods and properties of the problems they are designed to address. One possible extension of the work in characterizing networks consists of investigating other sampling models. For instance, there have been attempts to characterize the Internet router topology using traceroute sampling. This kind of sampling is based on the `traceroute` software tool, which allows a user to sample minimum-cost paths from controlled hosts to random hosts on the Internet. Estimating structural properties on arbitrary graphs using traceroute is an important and well-known hard problem [1, 23] and it remains open to date. In [64] we have shed some light on what is attainable when the graph comes from a branching process from an empirical perspective.

There are still several open questions regarding selective harvesting over networks. The proposed algorithm, D^3TS, greedily selects the node to be queried at each step. Ideally, we would like to query the nodes more likely to lead to the greatest number of targets. While the fact that the network is only partially observed prevent us from applying the same ideas from networked active search to account for future impact (see Section 5.8.1), increasing the size of the fringe set can be helpful as this gives the search algorithm more options to query. One way to achieve this goal is by estimating the degree of each fringe node and

selecting, among those that are most likely to be targets, those that have the largest expected degrees.

Another question left as future work concerns the use of context to guide the arm's choice in D^3TS. Currently, the context is only used to choose a node after the arm (classifier) is selected. Presumably, using context to select an arm could help in situations where one classifier predicts a certain action has a very high reward with small error margin.

Another avenue of investigation consists of pursuing a systematic way of achieving diversity in selective harvesting. In our approach, diversity is infused through the use of different models. An open question is whether multiple instances of the same model can achieve diversity by setting their parameters differently or by assigning different weights to the observations used to fit each instance. This investigation requires defining measure of diversity that is correlated with the performance of the search algorithm.

While we show that classifier diversity can severely increase the number of targets found in selective harvesting, it is not clear that this is the only mechanism that can mitigate the tunnel vision effect. We investigated the effect of sampling nodes probabilistically by mapping the node ranking computed by a given classifier to a distribution. While this showed no significant improvement over deterministic sampling, this approach could be further investigated by taking node scores into account when defining a distribution over possible choices.

Last, we lack a good understanding of what causes certain standalone classifiers to perform well on a given dataset. Investigating which features of a network have positive (or negative) correlation with the performance of a given classifier can be useful to select the set of classifiers to be combined through D^3TS, or to propose an improved solution that accounts for how efficient each classifier in this set is on the specific dataset at hand.

APPENDIX A

HYBRID ESTIMATOR AND ITS STATISTICAL PROPERTIES

First, we derive the recursive variant of the hybrid estimator. From that we derive its non-recursive variant. Next, we show that the non-recursive variant is asymptotically unbiased. In the case of undirected networks where the average degree is given, we show that the resulting hybrid estimator of the undirected degree mass is the minimum variance unbiased estimator (MVUE).

Let us recall variables and constants used in the definition of the hybrid estimator:

n_i	number of vertex samples with label i
$\theta_{i,j}$	fraction of nodes in $G^{(t)}$ with label i and undirected degree j
$m_{i,j}$	number of edge samples with label i and undirected degree j
$m_i = \sum_j m_{i,j}$	total number of edge samples with label i
$N = \sum_i n_i$	total number of vertex samples
$M = \sum_i m_i$	total number of edge samples
$B = N + M$	total budget

We approximate random walk samples in DUFS by uniform edge samples from G_u. Experience from previous papers shows us that this approximation works very well in practice. This yields the following likelihood function

$$L(\boldsymbol{\theta}|\mathbf{n},\mathbf{m}) = \frac{\prod_i \theta_i^{n_i} \prod_k ((w+k)\theta_{i,k})^{m_{i,k}}}{\left(\sum_{s,t}(w+t)\theta_{s,t}\right)^M}. \tag{A.1}$$

The key idea in our derivation is that we can bypass the numerical estimation of the $\theta_{i,j}$'s by noticing that $\theta_{i,j} \propto \theta_i$, $\theta_{i,j} \propto m_{i,j}$ and $\theta_{i,j} \propto 1/(w+j)$. Hence, the maximum

likelihood estimator of $\theta_{i,j}$ for $j = 1, \ldots, Z$ is the Hansen-Hurwitz estimator

$$\hat{\theta}_{i,j} = \frac{\theta_i m_{i,j}}{(w+j)\mu_i}, \tag{A.2}$$

where $\mu_i = \sum_k m_{i,k}/(w+k)$.

Substituting (A.2) in (A.1) yields

$$L(\boldsymbol{\theta}|\mathbf{n},\mathbf{m}) = \frac{\prod_i \theta_i^{n_i} \prod_k (\theta_i m_{i,k}/\mu_i)^{m_{i,k}}}{\left(\sum_s \theta_s \sum_z m_{s,z}/\mu_s\right)^M}. \tag{A.3}$$

The log-likelihood approximation is then given by

$$\mathcal{L}(\boldsymbol{\theta}|\mathbf{n},\mathbf{m}) = -M \log\left(\sum_s \theta_s \sum_z \frac{m_{s,z}}{\mu_s}\right) + \sum_i n_i \log\theta_i + \sum_k m_{i,k}(\log\theta_i + \log m_{i,k} - \log\mu_i). \tag{A.4}$$

We rewrite θ_i as $e^{\beta_i}/\sum_j e^{\beta_j}$ to account for the distribution constraints $\sum_i \theta_i = 1$ and $\theta_i \in [0,1]$. Hence, we have

$$\mathcal{L}(\boldsymbol{\beta}|\mathbf{n},\mathbf{m}) = -M \log\left(\sum_s \frac{e^{\beta_s} m_s}{\mu_s}\right) + \sum_i (n_i + m_i)\beta_i - N \log\left(\sum_j e^{\beta_j}\right) + C, \tag{A.5}$$

where $m_i = \sum_k m_{i,k}$ and C is a constant that does not depend on β.

The partial derivative w.r.t. β_i is given by

$$\frac{\partial \mathcal{L}(\boldsymbol{\beta}|\mathbf{n},\mathbf{m})}{\partial \beta_i} = -\frac{M e^{\beta_i} m_i/\mu_i}{\sum_s e^{\beta_s} m_s/\mu_s} + n_i + m_i - \frac{N e^{\beta_i}}{\sum_j e^{\beta_j}}. \tag{A.6}$$

Setting $\partial \mathcal{L}(\boldsymbol{\beta}|\mathbf{n},\mathbf{m})/\partial\beta_i = 0$ and substituting back θ_i yields

$$\theta_i^\star = \frac{n_i + m_i}{N + M \frac{m_i/\mu_i}{\sum_s \theta_s^\star m_s/\mu_s}}. \tag{A.7}$$

Theorem A.1. *Let $N = \alpha B$ and $M = (1-\alpha)B$, for some $0 < \alpha < 1$. In the limit as $B \to \infty$,*

$$\hat{\theta}_i = \frac{n_i + m_i}{N + M\frac{m_i}{\mu_i \hat{d}}}, \tag{A.8}$$

where $\mu_i = \sum_k m_{i,k}/(w+k)$ and $\hat{d} = M/\sum_i \mu_i$, is an unbiased estimate of θ_i.

Proof. In the limit as $B \to \infty$, we have

$$E[n_i] = N\theta_i, \qquad E[m_{i,k}] = M\frac{(w+k)\theta_{i,k}}{\sum_{s,l}(w+l)\theta_{sl}}, \qquad E[m_i] = M\frac{\sum_k (w+k)\theta_{i,k}}{\sum_{s,l}(w+l)\theta_{s,l}},$$

and thus,

$$E[\mu_i] = M\frac{\sum_k (w+k)\theta_{i,k}/(w+k)}{\sum_{s,l}(w+l)\theta_{sl}} = M\frac{\theta_i}{\sum_{s,l}(w+l)\theta_{sl}} \quad \text{and} \quad E\left[\frac{m_i}{\mu_i}\right] = \frac{\sum_k (w+k)\theta_{i,k}}{\theta_i}.$$

It follows that

$$\lim_{B\to\infty} E[\hat{d}] \;=\; \frac{M}{M\frac{\sum_i \theta_i}{\sum_{s,l}(w+l)\theta_{sl}}} = \sum_{s,l}(w+l)\theta_{sl}.$$

Substituting the above in eq. (A.8), we have

$$\lim_{B\to\infty} E[\theta_i^\star] = \frac{N\theta_i + M\frac{\sum_k (w+k)\theta_{i,k}}{\sum_{s,l}(w+l)\theta_{s,l}}}{N + M\frac{\sum_k (w+k)\theta_{i,k}/\theta_i}{\sum_{s,l}(w+l)\theta_{s,l}}} = \theta_i.$$

This concludes the proof.

In Section 3.4.2.2 we mentioned a special case of the previous estimator, where the vertex label is the undirected degree itself. We prove that, when the average degree $\sum_j j\theta_j$ is known, this estimator is the minimum variance unbiased estimator (MVUE) of θ_i.

Theorem A.2. *The quantity*

$$\bar{\theta}_i = \frac{n_i + m_i}{N + M(w+i)/\bar{\mu}},$$

where $\bar{\mu} = w + \sum_j j\theta_j$*, is an unbiased estimate of* θ_i*.*

Proof. We know that $n_i \sim \text{Binomial}(N, \theta_i)$ and $m_i \sim \text{Binomial}(M, (w+i)\theta_i/\bar{\mu})$. Hence,

$$\begin{aligned}
E[\hat{\theta}_i] &= \sum_{n_i, m_i} \left[\frac{n_i + m_i}{N + M(w+i)/\bar{\mu}} \overbrace{\binom{N}{n_i} \theta_i^{n_i} (1-\theta_i)^{N-n_i}}^{A(n_i)} \times \right. \\
&\qquad \left. \overbrace{\binom{M}{m_i} \left(\frac{(w+i)\theta_i}{\bar{\mu}}\right)^{m_i} \left(1 - \frac{(w+i)\theta_i}{\bar{\mu}}\right)^{M-m_i}}^{B(m_i)} \right] \\
&= \frac{1}{N + M(w+i)/\bar{\mu}} \left(\sum_{n_i} n_i A(n_i) \sum_{m_i} B(m_i) + \sum_{m_i} m_i B(m_i) \sum_{n_i} A(n_i) \right) \\
&= \frac{1}{N + M(w+i)/\bar{\mu}} \left(\sum_{n_i} n_i A(n_i) + \sum_{m_i} m_i B(m_i) \right) \\
&= \frac{1}{N + M(w+i)/\bar{\mu}} \left(N\theta_i + M(w+i)\theta_i/\bar{\mu} \right) \\
&= \theta_i.
\end{aligned}$$

□

Having proved that $\hat{\theta}_i$ is unbiased, we are now ready to show that it is also the minimum variance unbiased estimator (MVUE). In order to do so, we prove Lemmas A.3 and A.5 that show respectively that $n_i + m_i$ is a sufficient and complete statistic of θ_i.

Lemma A.3. *The statistic* $n_i + m_i$ *is a sufficient statistic w.r.t the likelihood of* θ_i*.*

Proof. The log-likelihood equation for estimator (3.8) is given by

$$\begin{aligned} L(\boldsymbol{\theta}|\mathbf{n},\mathbf{m}) &= \frac{\prod_i \theta_i^{n_i} \prod_j ((w+j)\theta_j)^{m_j}}{\hat{\mu}^M} \\ &= \frac{\prod_j (w+j)^{m_j}}{\hat{\mu}^M} \prod_i \theta_i^{n_i+m_i}. \end{aligned} \tag{A.9}$$

We can see from eq. (A.9) that the likelihood function $L(\boldsymbol{\theta}|\mathbf{n},\mathbf{m})$ can be factored into a product such that one factor, $\prod_j (w+j)^{m_j}/\hat{\mu}^M$, does not depend on θ_i and the other factor, which does depend on θ_i, depends on $\mathbf{n}$ and $\mathbf{m}$ only through $n_i + m_i$. From the Fisher-Neyman factorization Theorem [51], we conclude that $n_i + m_i$ is a sufficient statistic for the distribution of the sample.

□

We now prove that $n_i + m_i$ is also a complete statistic for the distribution of the sample.

Definition A.4. *Let X be a random variable whose probability distribution belongs to a parametric family of probability distributions P_θ parametrized by θ. The statistic s is said to be complete for the distribution of X if for every measurable function g (which must be independent of θ) the following implication holds:*

$$E(g(s(X))) = 0 \textit{ for all } \theta \Rightarrow P_\theta(g(s(X)) = 0) = 1 \textit{ for all } \theta.$$

Lemma A.5. *The statistic $n_i + m_i$ is a complete statistic w.r.t. the likelihood of θ_i.*

Proof.

$$\begin{aligned} E[g(n_i+m_i)] &= 0 \\ \sum_{n_i,m_i} g(n_i+m_i) P_\theta(n_i,m_i) &= 0 \\ \sum_{n_i,m_i} g(n_i+m_i) A(n_i) B(m_i) &= 0 \end{aligned} \tag{A.10}$$

The LHS of (A.10) is a polynomial of degree $M + N$ on θ_i. Hence, it can be written as

$$C_0 + C_1\theta_i + C_2\theta_i^2 + \ldots + C_{N+M}\theta_i^{N+M} = 0. \tag{A.11}$$

We prove that $P_\theta(g(s(X)) = 0) = 1$ for all θ by contradiction. Suppose that there is a θ such that $P_\theta(g(s(X)) \neq 0) > 0$. In order to have $E(g(s(X))) = 0$, there must be terms for which $g(.)$ is strictly positive and terms for which $g(.)$ is strictly negative. Let $g(h_1)$ be the smallest h_1 such that $g(h_1) > 0$. Let $g(h_2)$ be the smallest h_2 such that $g(h_2) < 0$. Let $h = \min(h_1, h_2)$.

Expanding $A(n_i)B(m_i)$ in eq. (A.10) we note that the degree of the resulting polynomial is at least $n_i + m_i$ on θ_i. Therefore, the coefficient C_h in eq. (A.11) associated with θ_i^h cannot have terms of $g(.)$ larger than h. Then C_h can only be zero if $h_1 = h_2$ which is a contradiction.

□

Theorem A.6. *The unbiased estimator $\bar{\theta}_i$ is the minimum variance unbiased estimator (MVUE) of θ_i.*

Proof. According to the Lehmann-Scheffe Theorem [51], if $T(\mathbb{S})$ is a complete sufficient statistic, there is at most one unbiased estimator that is a function of $T(\mathbb{S})$. From Lemmas A.3 and A.5, we have that $n_i + m_i$ is a complete sufficient statistic of θ_i. Clearly, the unbiased estimator $\hat{\boldsymbol{\theta}}$ in eq. (A.8) is a function $n_i + m_i$. Therefore, $\hat{\theta}_i$ must be the MVUE.

□

APPENDIX B

SET SIZE DISTRIBUTION PROOFS

Let $B(p) = [b_{ji}(p)], j, i = 1, \ldots, W$ be a matrix whose elements are given by

$$b_{ji}(p) \equiv P\left[\alpha(\mathcal{S}) = j \,|\, \alpha(\mathcal{S}) > 0, |\mathcal{S}| = i\right] = \frac{\binom{i}{j} p^j q^{i-j}}{1 - q^i}, \quad \text{if } 0 < j \leq i, \tag{B.1}$$

and $b_{ij}(p) = 0$ otherwise, where $q = 1 - p$.

Lemma B.1 shows a closed formula for the inverse of $B(p)$.

Lemma B.1. *$B(p)^{-1} = [b^{\star}_{ji}(p)]$ $(i, j = 1, \ldots, W)$, where*

$$b^{\star}_{ji}(p) = \begin{cases} \binom{i}{j} p^{-i} (-q)^{i-j} (1 - q^j) & i \geq j \\ 0 & i < j. \end{cases}$$

Proof. Let $B(p)^{-1} = [b^{\star}_{ji}(p)]$ with $b^{\star}_{ji}(p)$ defined above. We first show that $Y = B(p)B(p)^{-1}$ is an identity matrix. Consider element (j, i) of Y:

$$y_{ji} = \sum_{l=1}^{W} b_{jl}(p) b^{\star}_{li}(p)\,. \tag{B.2}$$

We have three cases: $j > i$, $j = i$, and $j < i$.

Case 1, $j > i$: eq. (B.2) yields $y_{ji} = 0$ since $b_{jl}(p) = 0, \forall l \leq i$ and $b^{\star}_{li}(p) = 0, \forall l > i$.

Case 2, $j = i$: Here $b_{jl}(p)b^{\star}_{lj}(p) = 0, \forall l \neq j$ and (B.2) yields

$$y_{jj} = \frac{p^j}{1-q^j} \cdot p^{-j}(1-q^j) = 1\,.$$

Case 3, $j < i$: eq. (B.2) yields

$$\begin{aligned} y_{ji} &= \sum_{l=j}^{i} (-1)^{i-l} p^{j-i} q^{i-j} \binom{l}{j} \binom{i}{l} \\ &= p^{j-i} q^{i-j} \sum_{l=j}^{i} (-1)^{i-l} \binom{i}{j} \binom{i-j}{l-j} \\ &= p^{j-i} q^{i-j} \binom{i}{j} \sum_{l=j}^{i} (-1)^{i-l} \binom{i-j}{l-j} \\ &= p^{j-i} q^{i-j} \binom{i}{j} (1-1)^{i-j} \\ &= 0 \end{aligned}$$

Thus, $y_{jj} = 1, \forall j$ and $y_{ji} = 0, \forall j \neq i$, which concludes our proof. □

Lemma B.1 directly yields the inverse of the Fisher information matrix $J^{(\phi)}$ of a single observed set, as seen in the following lemma.

Lemma B.2. $(J^{(\phi)})^{-1} = [[(J^{(\phi)})^{-1}]_{ij}]$ $(i, j = 1, 2, \ldots, W)$*, where*

$$[(J^{(\phi)})^{-1}]_{ij} = \sum_{k=\max(i,j)}^{W} \left(\frac{q}{p}\right)^{2k} \binom{k}{j} \binom{k}{i} (-1)^{-i-j} (q^{-i}-1)(q^{-j}-1) d_k(\boldsymbol{\theta}) \qquad \text{(B.3)}$$

Proof. Denote $R^{(\boldsymbol{\phi})}(p) = [R^{(\boldsymbol{\phi})}_{ji}(p)] = B^{-1}(p)\text{diag}(B(p)\boldsymbol{\phi})^{-1}$, where $R^{(\boldsymbol{\phi})}_{ji}(p) = b^{\star}_{ji}(p)d_i(\boldsymbol{\phi})$. Based on Lemma B.1 and eq. (4.2), we have

$$R^{(\boldsymbol{\phi})}_{ji}(p) = \begin{cases} \binom{i}{j} p^{-i}(-q)^{i-j}(1-q^j) d_i(\boldsymbol{\phi}), & i \geq j, \\ 0, & i < j. \end{cases} \qquad \text{(B.4)}$$

Since $J^{(\phi)} = R^{(\boldsymbol{\phi})}(p)(B(p)^{-1})^{\mathsf{T}}$, $[(J^{(\phi)})^{-1}]_{ji}$ is computed as the following equation based on Lemma B.1 and eq. (B.4)

$$\begin{aligned}
[(J^{(\phi)})^{-1}]_{ji} &= \sum_{k=1}^{W} R_{jk}^{(\boldsymbol{\phi})}(p) b_{ik}^{\star}(p) \\
&= \sum_{k=\max(i,j)}^{W} \frac{\binom{k}{j}\binom{k}{i}(-q)^{2k-i-j}(1-q^i)(1-q^j)d_k(\boldsymbol{\phi})}{p^{2k}} \\
&= \sum_{k=\max(i,j)}^{W} \left(\frac{q}{p}\right)^{2k} \binom{k}{j}\binom{k}{i}(-1)^{-i-j}(q^{-i}-1)(q^{-j}-1)d_k(\boldsymbol{\phi})
\end{aligned}$$

□

Lemma B.3. $(J^{(\theta)})^{-1} = [[(J^{(\theta)})^{-1}]_{ij}]$ $(i, j = 1, 2, \ldots, W)$, *where*

$$[(J^{(\theta)})^{-1}]_{ii} = \frac{1}{\eta^2}\left(\frac{[(J^{(\phi)})^{-1}]_{ii}}{(1-q^i)^2} + \theta_i^2 \sum_{j=1}^{W}\sum_{k=1}^{W} \frac{[(J^{(\phi)})^{-1}]_{kj}}{(1-q^k)(1-q^j)} - 2\theta_i \sum_{j=1}^{W} \frac{[(J^{(\phi)})^{-1}]_{ij}}{(1-q^i)(1-q^j)}\right) \tag{B.5}$$

where $\eta = \sum_{i=1}^{W} \phi_i/(1-q^i)$.

Proof. The relationship between $(J^{(\theta)})^{-1}$ and $(J^{(\boldsymbol{\phi})})^{-1}$ is given by

$$(J^{(\theta)})^{-1} = \nabla H (J^{(\boldsymbol{\phi})})^{-1} \nabla H^{\mathsf{T}}, \tag{B.6}$$

where $\nabla H = [h_{ik}]$ with $h_{ik} = \partial\theta_k(\boldsymbol{\phi})/\partial\phi_i$. Hence

$$h_{ik} = \begin{cases} -\frac{\phi_i/(\eta(1-q^i))}{\eta(1-q^k)} & i \neq k \\ \frac{1-\phi_i/(\eta(1-q^i))}{\eta(1-q^i)} & i = k \end{cases}$$

where $\eta = \sum_{k=1}^{W} \phi_k/(1-q^k)$ is a constant. Note that from eq. (4.3) we have $\theta_i = \phi_i/(\eta(1-q^i))$. Therefore the diagonal elements of $(J^{(\theta)})^{-1}$ can be written as

$$
\begin{aligned}
[(J^{(\theta)})^{-1}]_{ii} &= \sum_{j=1}^{W}\sum_{k=1}^{W} h_{ik}[(J^{(\phi)})^{-1}]_{kj}h_{ij}^{\mathsf{T}} \\
&= \sum_{\substack{j=1\\ j\neq i}}^{W}\sum_{\substack{k=1\\ k\neq i}}^{W}\left(-\frac{\theta_i}{\eta(1-q^k)}\right)[(J^{(\phi)})^{-1}]_{kj}\left(-\frac{\theta_i}{\eta(1-q^j)}\right) + \\
&\quad \sum_{\substack{j=1\\ j\neq i}}^{W}\left(\frac{1-\theta_i}{\eta(1-q^i)}\right)[(J^{(\phi)})^{-1}]_{ij}\left(-\frac{\theta_i}{\eta(1-q^j)}\right) + \\
&\quad \sum_{\substack{k=1\\ k\neq i}}^{W}\left(-\frac{\theta_i}{\eta(1-q^k)}\right)[(J^{(\phi)})^{-1}]_{ki}\left(\frac{1-\theta_i}{\eta(1-q^i)}\right) + \left(\frac{1-\theta_i}{\eta(1-q^i)}\right)^2[(J^{(\phi)})^{-1}]_{ii} \\
&= \frac{1}{\eta^2}\left(\frac{[(J^{(\phi)})^{-1}]_{ii}}{(1-q^i)^2} + \theta_i^2\sum_{j=1}^{W}\sum_{k=1}^{W}\frac{[(J^{(\phi)})^{-1}]_{kj}}{(1-q^k)(1-q^j)} - 2\theta_i\sum_{j=1}^{W}\frac{[(J^{(\phi)})^{-1}]_{ij}}{(1-q^i)(1-q^j)}\right) \qquad \text{(B.7)}
\end{aligned}
$$

□

We split eq. (B.5) in three parts to carry out its analysis:

$$
[(J^{(\theta)})^{-1}]_{ii} = \frac{1}{\eta^2}\left(\underbrace{\frac{[(J^{(\theta)})^{-1}]_{ii}}{(1-q^i)^2}}_{A_1(i)} + \underbrace{\theta_i^2\sum_{j=1}^{W}\sum_{k=1}^{W}\frac{[(J^{(\theta)})^{-1}]_{kj}}{(1-q^k)(1-q^j)}}_{A_2(j)} - \underbrace{2\theta_i\sum_{j=1}^{W}\frac{[(J^{(\theta)})^{-1}]_{ij}}{(1-q^i)(1-q^j)}}_{A_3(i)}\right). \qquad \text{(B.8)}
$$

Analysis of $A_1(i)$

Based on Lemma B.2 and eq. (4.2), we have

Lemma B.4.

$$
A_1(i) = \eta q^{-2i}\sum_{j=0}^{W-i}\binom{i+j}{i}q^{j+i}\theta_{j+i}g_{ij}. \qquad \text{(B.9)}
$$

where $\eta = \sum_{k=1}^{W}\phi_k/(1-q^k)$ *and* $g_{ij} = \sum_{k=0}^{j}\binom{i+k}{i}\binom{j}{k}(q/p)^{k+i}$.

Proof.

$$
[(J^{(\phi)})^{-1}]_{ii} = \sum_{k=i}^{W}\left(\frac{q}{p}\right)^{2k}\binom{k}{i}^2(-1)^{-2i}(q^{-i}-1)^2 d_k(\boldsymbol{\phi})
$$

$$
\begin{aligned}
&= \sum_{k=i}^{W}\sum_{j=k}^{W}\left(\frac{q}{p}\right)^{2k}\binom{k}{i}^2(-1)^{-2i}(q^{-i}-1)^2\frac{\binom{j}{k}p^k q^{j-k}\phi_j}{1-q^j} \\
&= (q^{-i}-1)^2\sum_{j=i}^{W}\binom{j}{i}\frac{q^j\phi_j}{1-q^j}\sum_{k=i}^{j}\binom{k}{i}\binom{j-i}{k-i}(q/p)^k \\
&= (q^{-i}-1)^2\sum_{j=0}^{W-i}\binom{i+j}{i}\frac{q^{i+j}\phi_{i+j}g_{ij}}{1-q^{i+j}} \qquad \text{(B.10)}
\end{aligned}
$$

where $g_{ij}=\sum_{k=0}^{j}\binom{i+k}{i}\binom{j}{k}(q/p)^{i+k}$.

Since $\phi_i/(1-q^i)=\theta_i\cdot\eta$, we can eq. (B.9) as a function of θ:

$$
[(J^{(\phi)})^{-1}]_{ii}=\eta\left(q^{-i}-1\right)^2\sum_{j=0}^{W-i}\binom{i+j}{i}q^{i+j}\theta_{i+j}g_{ij}.
$$

Therefore

$$
A_1(i)=\eta q^{-2i}\sum_{j=0}^{W-i}\binom{i+j}{i}q^{i+j}\theta_{i+j}g_{ij}. \qquad \text{(B.11)}
$$

□

Lemma B.5. *We have the following bounds for $A_1(i)$:*

$$
A_1(i)<C_i\sum_{k=0}^{i}c_{ik}\sum_{j=0}^{\infty}\mathbf{1}\{k\le j\}(i+j)^{2i}\left(\frac{q}{p}\right)^{i+j}\theta_{i+j} \qquad \text{(B.12)}
$$

and

$$
A_1(i)>C_i c_{ii}\sum_{j=i(i-1)}^{W-i}j^{2i}\left(\frac{q}{p}\right)^{i+j}\theta_{i+j} \qquad \text{(B.13)}
$$

where

$$
C_i=\frac{\eta q^{-i}}{(i!)^2}
$$

and

$$
c_{ik}=\binom{i}{k}q^k\prod_{l=0}^{i-k-1}(i-l),\quad k=0,\ldots,i;i=1,\ldots W.
$$

Proof. Since the i-th derivative of $(q/p)^{i+k}$ with respect to q/p, is

$$\frac{\mathrm{d}^i(q/p)^{i+k}}{\mathrm{d}(q/p)^i} = \prod_{l=1}^{i}(k+l)(q/p)^k,$$

we have the following equations for g_{ij}

$$\begin{aligned} g_{ij} &= \frac{1}{i!}\left(\frac{q}{p}\right)^i \sum_{k=0}^{j}\prod_{l=1}^{i}(k+l)\binom{j}{k}(q/p)^k \\ &= \frac{1}{i!}\left(\frac{q}{p}\right)^i \sum_{k=0}^{j}\binom{j}{k}\frac{\mathrm{d}^i(q/p)^{i+k}}{\mathrm{d}(q/p)^i} \\ &= \frac{1}{i!}\left(\frac{q}{p}\right)^i \frac{\mathrm{d}^i\left(\sum_{k=0}^{j}\binom{j}{k}(q/p)^{i+k}\right)}{\mathrm{d}(q/p)^i} \\ &= \frac{1}{i!}\left(\frac{q}{p}\right)^i \frac{\mathrm{d}^i\left((q/p)^i(1+q/p)^j\right)}{\mathrm{d}(q/p)^i}. \end{aligned}$$

Using a general form of the product rule [67, pp. 318] yields

$$g_{ij} = \frac{1}{i!}\left(\frac{q}{p}\right)^i \sum_{k=0}^{\min\{i,j\}}\binom{i}{k}\left(\frac{1}{p}\right)^{j-k}\prod_{l=0}^{k-1}(j-l)\left(\frac{q}{p}\right)^k \prod_{l=0}^{i-k-1}(i-l), \tag{B.14}$$

where to simplify the expression we define $\prod_{l=0}^{-1}\cdots = 1$.

Substituting (B.14) back into (B.11), we obtain the following expression for $A_1(i)$

$$A_1(i) = C_i \sum_{k=0}^{i} c_{ik} \sum_{j=0}^{W-i} \mathbf{1}\{k \le j\} \prod_{l=1}^{i}(j+l) \prod_{l=0}^{k-1}(j-l)(q/p)^{i+j}\theta_{i+j} \tag{B.15}$$

where

$$C_i = \frac{\eta q^{-i}}{(i!)^2}$$

and

$$c_{ik} = \binom{i}{k} q^k \prod_{l=0}^{i-k-1}(i-l), \quad k = 0, \ldots, i; i = 1, \ldots, W.$$

We have the following upper bounds for $A_1(i)$,

$$\begin{aligned} A_1(i) &< C_i \sum_{k=0}^{i} c_{ik} \sum_{j=0}^{W-i} \mathbf{1}\{k \le j\}(i+j)^{2i} \big(\frac{q}{p}\big)^{i+j} \theta_{i+j} && \text{(B.16)} \\ &< C_i \sum_{k=0}^{i} c_{ik} \sum_{j=0}^{\infty} \mathbf{1}\{k \le j\}(i+j)^{2i} \big(\frac{q}{p}\big)^{i+j} \theta_{i+j}. && \text{(B.17)} \end{aligned}$$

A lower bound is obtained by noting that

$$\begin{aligned} \prod_{l=1}^{i}(j+l) \prod_{l=0}^{k-1}(j-l) &> j^{i-k} \prod_{l=1}^{k}(j+l) \prod_{l=1}^{k}(j-l+1) \\ &= j^{i-k} \prod_{l=1}^{k}(j^2+j+l-l^2). \end{aligned}$$

The latter is greater than or equal to j^{2i} whenever $j > i(i-1)$ yielding

$$A_1(i) > C_i c_{ii} \sum_{j=i(i-1)}^{W-i} j^{2i} \big(\frac{q}{p}\big)^{i+j} \theta_{i+j}. \qquad \text{(B.18)}$$

□

Analysis of $A_2(i)$

$$\begin{aligned} \sum_{i=1}^{W}\sum_{j=1}^{W} \frac{[(J^{(\theta)})^{-1}]_{ij}}{(1-q^i)(1-q^j)} &= \sum_{i=1}^{W}\sum_{j=1}^{W}\sum_{k=1}^{W} \frac{\binom{k}{j}\binom{k}{i}\left(\frac{q}{p}\right)^{2k}(-1)^{-j-i}(q^{-j}-1)(q^{-i}-1)d_k(\boldsymbol{\phi})}{(1-q^j)(1-q^i)} \\ &= \sum_{k=1}^{W}\left(\frac{q}{p}\right)^{2k} d_k(\boldsymbol{\phi}) \sum_{i=1}^{k}\sum_{j=1}^{k}\binom{k}{j}\binom{k}{i}(-q)^{-j-i} \\ &= \sum_{k=1}^{W}\left(\frac{q}{p}\right)^{2k} d_k(\boldsymbol{\phi}) \left(\sum_{i=1}^{k}\binom{k}{i}(-q)^{-i}\right)^2 \\ &= \sum_{k=1}^{W}\left(\frac{q}{p}\right)^{2k} d_k(\boldsymbol{\phi}) \left(\left(-\frac{q}{p}\right)^{-k}-1\right)^2 \qquad \text{using (G.2)} \end{aligned}$$

$$
\begin{aligned}
&= \sum_{k=1}^{W} d_k(\boldsymbol{\phi}) - 2\sum_{k=1}^{W}\left(-\frac{q}{p}\right)^k d_k(\boldsymbol{\phi}) + \sum_{k=1}^{W}\left(\frac{q}{p}\right)^{2k} d_k(\boldsymbol{\phi}) \\
&= 1 - 2\sum_{k=1}^{W}\left(-\frac{q}{p}\right)^k d_k(\boldsymbol{\phi}) + \sum_{k=1}^{W}\left(\frac{q}{p}\right)^{2k} d_k(\boldsymbol{\phi}). \qquad \text{(B.19)}
\end{aligned}
$$

First, note that

$$
\begin{aligned}
\sum_{k=1}^{W}\left(-\frac{q}{p}\right)^k d_k(\boldsymbol{\phi}) &= \sum_{k=1}^{W}\left(-\frac{q}{p}\right)^k \sum_{j=1}^{W}\binom{j}{k} p^k q^{j-k}\theta_j\eta \\
&= \eta\sum_{j=1}^{W} q^j\theta_j \sum_{k=1}^{j}\binom{j}{k}(-1)^k \\
&= -\eta\sum_{j=1}^{W} q^j\theta_j. \qquad \text{using (G.4)} \qquad \text{(B.20)}
\end{aligned}
$$

Also,

$$
\begin{aligned}
\sum_{k=1}^{W}\left(\frac{q}{p}\right)^{2k} d_k(\boldsymbol{\phi}) &= \sum_{k=1}^{W}\left(\frac{q}{p}\right)^{2k} \sum_{j=1}^{W}\binom{j}{k} p^k q^{j-k}\theta_j\eta \\
&= \eta\sum_{j=1}^{W} q^j\theta_j \sum_{k=1}^{j}\binom{j}{k}\left(\frac{q}{p}\right)^k \\
&= \eta\sum_{j=1}^{W} q^j\theta_j \left(\left(\frac{1}{p}\right)^j - 1\right) \qquad \text{using (G.3)} \\
&= \eta\left(\sum_{j=1}^{W}\left(\frac{q}{p}\right)^j \theta_j - \sum_{j=1}^{W} q^j\theta_j\right). \qquad \text{(B.21)}
\end{aligned}
$$

Substituting eqs. (B.20) and (B.21) into (B.19) yields

$$
\begin{aligned}
\sum_{i=1}^{W}\sum_{j=1}^{W}\frac{[(J^{(\theta)})^{-1}]_{ij}}{(1-q^i)(1-q^j)} &= 1 + \eta\left(2\sum_{j=1}^{W} q^j\theta_j + \sum_{j=1}^{W}\left(\frac{q}{p}\right)^j \theta_j - \sum_{j=1}^{W} q^j\theta_j\right) \qquad \text{(B.22)} \\
&= 1 + \eta\left(\sum_{j=1}^{W} q^j\theta_j + \sum_{j=1}^{W}\left(\frac{q}{p}\right)^j \theta_j\right). \qquad \text{(B.23)}
\end{aligned}
$$

Therefore,

$$A_2(i) = \theta_i^2 \left(1 + \eta \left(\sum_{j=1}^{W} q^j \theta_j + \sum_{j=1}^{W} \left(\frac{q}{p} \right)^j \theta_j \right) \right). \tag{B.24}$$

Note that $A_2(i)$ is positive and may diverge or not depending on the summation $\sum_{j=1}^{W} \left(\frac{q}{p} \right)^j \theta_j$.

Analysis of $A_3(i)$

Note that

$$\begin{aligned}
\sum_{k=1}^{W} \binom{k}{i} \left(-\frac{q}{p} \right)^k d_k(\boldsymbol{\phi}) &= \sum_{k=i}^{W} \binom{k}{i} \left(-\frac{q}{p} \right)^k \sum_{j=1}^{W} \binom{j}{k} p^k q^{j-k} \theta_j \eta \\
&= \eta \sum_{k=i}^{W} (-1)^k \sum_{j=1}^{W} \binom{j}{i} \binom{j-i}{k-i} q^j \theta_j \\
&= \eta \sum_{j=i}^{W} \binom{j}{i} q^j \theta_j \sum_{k=i}^{j} \binom{j-i}{k-i} (-1)^k \\
&= (-1)^i \eta \sum_{j=i}^{W} \binom{j}{i} q^j \theta_j \sum_{k=0}^{j-i} \binom{j-i}{k} (-1)^k \\
&= (-q)^i \eta \theta_i. \qquad \text{using (G.5)}
\end{aligned} \tag{B.25}$$

We also have

$$\begin{aligned}
\sum_{k=1}^{W} \binom{k}{i} \left(\frac{q}{p} \right)^{2k} d_k(\boldsymbol{\phi}) &= \sum_{k=1}^{W} \binom{k}{i} \left(\frac{q}{p} \right)^{2k} \sum_{j=1}^{W} \binom{j}{k} p^k q^{j-k} \theta_j \eta \\
&= \eta \sum_{k=1}^{W} \left(\frac{q}{p} \right)^k \sum_{j=1}^{W} \binom{j}{i} \binom{j-i}{k-i} q^j \theta_j \\
&= \eta \sum_{j=i}^{W} \binom{j}{i} q^j \theta_j \sum_{k=i}^{j} \binom{j-i}{k-i} \left(\frac{q}{p} \right)^k.
\end{aligned} \tag{B.26}$$

From eq. (B.25) and (B.26), we have

$$\sum_{j=1}^{W} \frac{[(J^{(\theta)})^{-1}]_{ij}}{(1-q^j)(1-q^i)} = \eta\theta_i - (-q)^{-i}\eta \sum_{j=i}^{W} \binom{j}{i} q^j \theta_j \sum_{k=i}^{j} \binom{j-i}{k-i} \left(\frac{q}{p}\right)^k \tag{B.27}$$

and hence,

$$A_3(i) = \underbrace{2\eta\theta_i^2}_{A_{3,1}(i)} \underbrace{- 2\theta_i(-q)^{-i}\eta \sum_{j=0}^{W-i} \binom{i+j}{i} q^{i+j}\theta_{i+j} \sum_{k=0}^{j} \binom{j}{k} \left(\frac{q}{p}\right)^{k+i}}_{A_{3,2}(i)}. \tag{B.28}$$

Since $A_{3,1}(i)$ is always finite, we only need to compare the magnitude of $A_1(i)$ and $A_{3,2}(i)$. Since $\sum_{k=0}^{j} \binom{j}{k} \left(\frac{q}{p}\right)^{k+i} < g_{ij}$, we can bound $|A_{3,2}(i)|$ by

$$|A_{3,2}(i)| \leq 2\theta_i q^{-i}\eta \sum_{j=0}^{W-i} \binom{i+j}{i} q^{i+j}\theta_{i+j} g_{ij}.$$

Therefore

$$A_1(i) - |A_{3,2}(i)| \geq (q^{-2i} - 2\theta_i q^{-i})\eta \sum_{j=0}^{W-i} \binom{i+j}{i} q^{i+j}\theta_{i+j} g_{ij}.$$

The RHS of the previous inequation is positive when

$$\begin{aligned} q^{-2i} &\geq 2\theta_i q^{-i} \\ \theta_i &\leq \frac{1}{2q^i} < \frac{1}{2}. \end{aligned}$$

Recall that we assumed that $\exists i_0$ such that $\theta_i \leq 1/2$ for all $i > i_0$. Thus by examining only $A_1(i)$ and $A_2(i)$ we can determine whether $[(J^{(\theta)})^{-1}]_{ii}$ diverges or not for $i > i_0$.

APPENDIX C

PROOF OF THEOREM 4.1

The lower bound of MSE$(T_i(\mathbb{S}))$, given by $[(J^{(\theta)})^{-1}]_{ii}$, is described for each of the three possible cases in Theorem 4.1. The corresponding proofs are shown in what follows.

1) When θ_W decreases faster than exponentially in W.

Proof. Suppose that θ_W decreases faster than exponentially in W. More precisely, assume that $-\log\theta_W = \omega(W)$. It follows that $\log(\theta_W/\theta_{W+1}) \to \infty$ as $W \to \infty$. Hence, for any $\epsilon > 0$, there exists a $W_0(\epsilon)$ such that $\log(\theta_W/\theta_{W+1}) > 1/\epsilon$ for $W > W_0(\epsilon)$. This implies $\theta_{W+1}/\theta_W < e^{-1/\epsilon}$ for $W > W_0(\epsilon)$. Given $p > 0$, we can choose ϵ such that $qe^{-1/\epsilon}/p < 1$. We now apply the ratio test for convergence of an infinite sum to each of the $i+1$ sums in the upper bound for $A_1(i)$ given by (B.12).

$$\frac{(W+i+1)^{2i}(q/p)^{W+i+1}\theta_{W+i+1}}{(W+i)^{2i}(q/p)^{W+i}\theta_{W+i}} < \frac{(W+i+1)^{2i}}{(W+i)^{2i}}\frac{qe^{-1/\epsilon}}{p}$$

for $W > W_0(\epsilon) - i$ and the latter expression becomes less than one as $W \to \infty$. Hence $A_1(i) = O(1)$ for $0 < p < 1$. A similar argument can be used to show that $A_2(i) = O(1)$. Hence, $[(J^{(\theta)})^{-1}]_{ii} = O(1)$ for $0 < p < 1$. □

2) When θ_W decreases exponentially in W.

Proof. Suppose that θ_W decreases exponentially in W. More precisely, let $\log\theta_W = W\log a + o(W)$ for $0 < a < 1$. Recall that $A_2(i)$ is positive. Therefore, the logarithm of $[(J^{(\theta)})^{-1}]_{ii}$ in (B.5) can be lower bounded as follows,

$$\log[(J^{(\theta)})^{-1}]_{ii} \geq \log A_1(i). \tag{C.1}$$

In addition, the logarithm of $A_1(i)$ in (B.9) can be bounded by

$$\begin{aligned}\log A_1(i) &\geq W\log(q/p) + \log\theta_W + o(W)\\ &= W\log(qa/p) + o(W)\end{aligned}$$

where the latter equality follows from the hypothesis. Now, if $qa/p > 1$, then $\log A_1(i) = \Omega(W)$, which implies $\log[(J^{(\theta)})^{-1}]_{ii} = \Omega(W)$. Note that $qa/p > 1$ iff $p < a/(a+1)$.

When $p = a/(a+1)$, then $qa/p = 1$. Hence the lower bound of $A_1(i)$ given by (B.13) is $\Omega(W^{2i+1})$. Hence, $[(J^{(\theta)})^{-1}]_{ii} = \Omega(W^{2i+1})$.

Similarly to the proof for the case where θ_W decreases faster than exponentially in W, we can use the ratio test for convergence of an infinite sum to show that for $qa/p < 1$, $A_1(i) = O(1)$. Hence, it follows that $[(J^{(\theta)})^{-1}]_{ii} = O(1)$ for $p > a/(a+1)$. □

3) When θ_W decreases slower than exponentially in W.

Proof. Suppose that θ_W decreases slower than exponentially in W. More precisely assume that $-\log\theta_W = o(W)$. The logarithm of $A_1(i)$ can be lower bounded as follows,

$$\begin{aligned}\log A_1(i) &\geq W\log(q/p) + \log\theta_W + o(W)\\ &= W\log(q/p) + o(W)\end{aligned}$$

The latter equality follows from the hypothesis. Now, if $q/p > 1$ (i.e., $p < 1/2$), then $\log A_1(i) \geq \Omega(W)$, which implies $\log[(J^{(\theta)})^{-1}]_{ii} = \Omega(W)$.

When $p \geq 1/2$, it follows that $A_2(i) = O(1)$. In particular if $p = 1/2$ and $\sum_{j=1}^{W} j^{2i}\theta_j = \omega(1)$, we can see from eq. (B.13) that $A_1(i) = \omega(1)$ and in turn, $[(J^{(\phi)})^{-1}]_{ii} = \omega(1)$.

Note that for $p = 1/2$ each of the $i+1$ sums in the upper bound for $A_1(i)$ given by (B.12) is bounded by the $2i$-th moment of the set size distribution. Hence, if $\sum_{j=1}^{W} j^{2i}\theta_j = O(1)$, then $[(J^{(\theta)})^{-1}]_{ii} = O(1)$.

Finally, when $p > 1/2$, an argument similar to that used in the case where θ_W decreases faster than exponentially yields $[(J^{(\theta)})^{-1}]_{ii} = O(1)$. □

APPENDIX D

SIMPLIFIED BOUNDS

It is worth noting that $A_2(i)$ gives us a lower bound on $[(J^{(\theta)})^{-1}]_{ii}$, as $A_1(i)-A_3(i) > 0$. Furthermore, the convergence of $A_2(i)$ is given by the convergence of the sum $\sum_{j=1}^{W}(q/p)^j\theta_j$. Therefore, we can write

$$[(J^{(\theta)})^{-1}]_{ii} = \Omega\left(\sum_{j=1}^{W}\left(\frac{1-p}{p}\right)^j\theta_j\right). \tag{D.1}$$

From that, we derive the following results.

1) When θ_W decreases faster than exponentially in W.

By definition, for any $\epsilon > 0$, there exists a $W_0(\epsilon)$ such that $\log(\theta_W/\theta_{W+1}) > 1/\epsilon$. Given $p > 0$, we can choose ϵ such that $qe^{-1/\epsilon}/p < 1$. The ratio test for convergence of an infinite sum reads

$$\frac{(q/p)^{j+1}\theta_{j+1}}{(q/p)^j\theta_j} < \frac{qe^{-1/\epsilon}}{p} \tag{D.2}$$

Let $a = qe^{-1/\epsilon}/p$. Hence, there exists a j^* such that for all $j > j^\star$, $((1-p)/p)^j\theta_j < a^j$, $j = 1, 2, \ldots$. Therefore, the sum converges to a constant for any $0 < p < 1$, yielding $[(J^{(\theta)})^{-1}]_{ii} = O(1)$.

2) When θ_W decreases exponentially in W.

By definition, there exists $0 < a < 1$ such that $\log\theta_W = W\log a + o(W)$. When $p \leq a/(a+1)$ it follows that $((1-p)/p)^j\theta_j \geq a^{-j}\theta_j = \Omega(1)$. Therefore, $[(J^{(\theta)})^{-1}]_{ii} = O(W)$. A tighter bound can be obtained by taking into account $A_1(i)$, yielding $\log[(J^{(\theta)})^{-1}]_{ii} = O(W)$ for $p < a/(a+1)$ and $[(J^{(\theta)})^{-1}]_{ii} = O(W^{2i+1})$ for $p = a/(a+1)$. On the other hand, for $p > a/(a+1)$, we have $((1-p)/p)^j\theta_j < a^j\theta_j = O(1)$. Hence, $[(J^{(\theta)})^{-1}] = O(1)$.

3) When θ_W decreases slower than exponentially in W.

When $p < 1/2$, it follows that $(1-p)/p = a > 1$. In this case, there exists a $j^\star$ such that for all $j > j^\star$, $((1-p)/p)^j\theta_j = a^j\theta_j = \Omega(1)$. Hence, $[(J^{(\theta)})^{-1}]_{ii} = O(W)$ for $p < 1/2$. Conversely, when $p > 1/2$, $(1-p)/p = a < 1$. Hence, there exists a $j^\star$ such that for all $j > j^\star$, $((1-p)/p)^j\theta_j = a^j\theta_j = O(1)$. Thus, $[(J^{(\theta)})^{-1}]_{ii} = O(1)$ for $p > 1/2$. At last, for $p = 1/2$, the summation is exactly 1, which also implies $[(J^{(\theta)})^{-1}]_{ii} = O(1)$. In the latter case (i.e., $p = 1/2$), a tigher bound is obtained by taking $A_1(i)$ into account, which yields $[(J^{(\theta)})^{-1}]_{ii} = \omega(1)$ if $\sum j = 1^W j^{2i}\theta_j = \omega(1)$ and $[(J^{(\theta)})^{-1}]_{ii} = O(1)$ if $\sum j = 1^W j^{2i}\theta_j = O(1)$.

APPENDIX E

ASYMPTOTIC EFFICIENCY AND ASYMPTOTIC NORMALITY OF THE MLE $T_i^*(\mathbb{S})$

In this section we show that there exists a Maximum Likelihood Estimator (MLE) $T_i^{(\phi)}(\mathbb{S})$ of ϕ_i that is asymptotic efficient (i.e., MSE$(T_i^*(\mathbb{S})) = [(J^{(\phi)})^{-1}]_{ii}$) and asymptotic normal. Since the Delta Method is an exact approximation for the Normal distribution, it follows that there exists a MLE $T_i^*(\mathbb{S})$ of θ_i that is asymptotic efficient, which can be obtained by applying the Delta Method to $T_i^{(\phi)}(\mathbb{S})$.

Consider the likelihood function obtained by expressing Eq. (4.2) as a function of $\boldsymbol{\phi}$:

$$d_j(\boldsymbol{\phi}) = \sum_{i=1}^{W} b_{ji}\phi_i.$$

From the sum-to-one contraint on the parameters, it follows that $\phi_1 = 1 - \sum_{i=2}^{W} \phi_i$. Thus we can rewrite the previous eq. as

$$d_j(\boldsymbol{\phi}) = b_{j1} + \sum_{i=2}^{W}(b_{ji} - b_{j1})\phi_i. \tag{E.1}$$

Hence,

$$\frac{\partial}{\partial \phi_k} \log d_j(\boldsymbol{\phi}) = \frac{b_{jk} - b_{j1}}{b_{j1} + \sum_{i=2}^{W}(b_{ji} - b_{j1})\phi_i} \qquad 2 < k < W.$$

From Theom. 5.1 [50, Chapter 5], we prove that there exists a MLE that is asymptotically efficient and asymptotically normal by showing that assumptions (A0)-(A2) and (A)-(D) are satisfied.

Proof. (A0) Follows from (E.1).

(A1) The support of ϕ_i for $2 \leq i \leq W$ is $0 < \phi_i < 1$ subject to $\sum_{i=2}^{W} \phi_i \leq 1$.

(A2) Observations are assumed to be independent.

(A3) Follows by the assumption that $0 < \phi_i < 1$ for $2 \leq i \leq W$.

(A) We have

$$\frac{\partial}{\partial \phi_k} d_j(\boldsymbol{\phi}) = b_{jk}, \quad 2 \leq k \leq W$$

and hence

$$\frac{\partial^3}{\partial \phi_m \partial \phi_l \partial \phi_k} d_j(\boldsymbol{\phi}) = 0, \quad 2 \leq k, l, m \leq W.$$

(B) The expectation of the first logarithmic derivative of f is

$$\begin{aligned} E_\phi \left[\frac{\partial}{\partial \phi_k} \log d_j(\boldsymbol{\phi}) \right] &= \sum_{j=1}^{W} \frac{b_{jk} - b_{j1}}{b_{j1} + \sum_{i=2}^{W} (b_{ji} - b_{j1}) \phi_i} \left(b_{j1} + \sum_{i=2}^{W} (b_{ji} - b_{j1}) \phi_i \right) \\ &= \sum_{j=1}^{W} b_{jk} - \sum_{j=1}^{W} b_{j1} \\ &= 1 - b_{11} \\ &= 0. \end{aligned}$$

As for the second derivative, we have

$$\begin{aligned} E \left[\frac{\partial}{\partial \phi_l} \log d_j(\boldsymbol{\phi}) \frac{\partial}{\partial \phi_k} \log d_j(\boldsymbol{\phi}) \right] &= \sum_{j=1}^{W} \frac{(b_{jl} - b_{j1})(b_{jk} - b_{j1})}{\left(b_{j1} + \sum_{i=2}^{W} (b_{ji} - b_{j1}) \phi_i \right)^2} \left(b_{j1} + \sum_{i=2}^{W} (b_{ji} - b_{j1}) \phi_i \right) \\ &= \sum_{j=1}^{W} \frac{(b_{jl} - b_{j1})(b_{jk} - b_{j1})}{b_{j1} + \sum_{i=2}^{W} (b_{ji} - b_{j1}) \phi_i}, \end{aligned}$$

which is equivalent to

$$E\left[-\frac{\partial^2}{\partial\phi_l\partial\phi_k}\log d_j(\boldsymbol{\phi})\right] = \sum_{j=1}^{W} -\left(-\frac{(b_{jk}-b_{j1})(b_{jl}-b_{j1})}{\left(b_{j1}+\sum_{i=2}^{W}(b_{ji}-b_{j1})\phi_i\right)^2}\left(b_{j1}+\sum_{i=2}^{W}(b_{ji}-b_{j1})\phi_i\right)\right)$$
$$= \sum_{j=1}^{W}\frac{(b_{jl}-b_{j1})(b_{jk}-b_{j1})}{b_{j1}+\sum_{i=2}^{W}(b_{ji}-b_{j1})\phi_i}.$$

(C) The vectors $\left[\frac{\partial}{\partial\phi_2}\log d_j(\boldsymbol{\phi}), \frac{\partial}{\partial\phi_3}\log d_j(\boldsymbol{\phi}), \ldots, \frac{\partial}{\partial\phi_W}\log d_j(\boldsymbol{\phi})\right]$ for $1 < j < W$ must be linearly independent with probability 1. Note that and $b_{jk} > 0 \iff j \leq k$ (in particular, $b_{j1} > 0 \iff j = 1$). It follows that for $j > k \geq 2$

$$\frac{\partial}{\partial\phi_k}\log d_j(\boldsymbol{\phi}) = \frac{b_{jk}-b_{j1}}{b_{j1}+\sum_{i=2}^{W}(b_{ji}-b_{j1})\phi_i} = 0, \qquad \text{for } j > k \geq 2, \text{ and}$$
$$\frac{\partial}{\partial\phi_k}\log d_j(\boldsymbol{\phi}) = \frac{b_{jk}}{\sum_{i=2}^{W}(b_{ji}-b_{j1})\phi_i} > 0, \qquad \text{for } j \leq k.$$

Therefore, the $j-1$ leftmost entries in the j-th vector are 0 while the remainder are positive. Hence the vectors are linearly independent.

(D) Consider a constant $\epsilon_j > 0$ such that $d_j(\boldsymbol{\phi}) = b_{j1} + \sum_{i=2}^{W}(b_{ji}-b_{j1})\phi_i \geq \epsilon_j$ for $1 \leq j \leq W$. Thus,

$$\left|\frac{\partial^3}{\partial\phi_m\partial\phi_l\partial\phi_k}d_j(\boldsymbol{\phi})\right| = \left|\frac{-(b_{jk}-b_{j1})(b_{jl}-b_{j1})\times 2(b_{jm}-b_{j1})\phi_m(b_{j1}+\sum_{i=2}^{W}(b_{ji}-b_{j1})\phi_i)}{\left(b_{j1}+\sum_{i=2}^{W}(b_{ji}-b_{j1})\phi_i\right)^4}\right|$$
$$= \left|\frac{2(b_{jk}-b_{j1})(b_{jd_j(\boldsymbol{\phi})l}-b_{j1})(b_{jm}-b_{j1})\phi_m}{\left(b_{j1}+\sum_{i=2}^{W}(b_{ji}-b_{j1})\phi_i\right)^3}\right|$$
$$\leq \left|\frac{2(b_{jk}-b_{j1})(b_{jl}-b_{j1})(b_{jm}-b_{j1})\phi_m}{\epsilon_j^3}\right|.$$

Since $M_{klm}(j) = \left|\frac{\partial^3}{\partial\phi_m\partial\phi_l\partial\phi_k}d_j(\boldsymbol{\phi})\right| < \infty$, then $E_\phi[M_{klm}(j)] < \infty$ for all k, l, m. $\square$

APPENDIX F

AVERAGE SET SIZE PROOFS

Lemma F.1. *Let p be the sampling probability and $\hat{m}_\phi$ denote an unbiased estimate of the average size of the observed sets m_ϕ. Then,*

$$\mathit{MSE}(\hat{m}_\phi) = O\left(\frac{m_\phi^{(2)} - m_\phi^2}{N}\right).$$

Proof. The estimation error lower bound of the average set size is [89, p. 83, Prop. 3]

$$\text{MSE}(\hat{m}_\phi) \geq \frac{(1,\ldots,W)(J^{(\phi)})^{-1}(1,\ldots,W)^{\mathrm{T}} - m_\phi^2}{N}. \tag{F.1}$$

Lemma B.2 yields

$$\begin{aligned}
& (1,\ldots,W)(J^{(\phi)})^{-1}(1,\ldots,W)^{\mathrm{T}} \\
= & \sum_{k=1}^{W}\sum_{i=1}^{k}\sum_{j=1}^{k} ij\binom{k}{j}\binom{k}{i}\left(\frac{q}{p}\right)^{2k}(-1)^{2k-i-j}(q^{-i}-1)(q^{-j}-1)d_k(\boldsymbol{\phi}) \\
= & \sum_{k=1}^{W}(q/p)^{2k}d_k(\boldsymbol{\phi})\left(\sum_{i=1}^{k} i\binom{k}{i}\frac{q^{-i}-1}{(-1)^i}\right)\left(\sum_{j=1}^{k} j\binom{k}{j}\frac{q^{-j}-1}{(-1)^j}\right) \\
= & d_1(\boldsymbol{\phi}) + \sum_{k=2}^{W}(q/p)^{2k}d_k(\boldsymbol{\phi})\left(\left(-\frac{1-q}{q}\right)^k\frac{k}{1-q}\right)^2 \\
= & \left(1-\frac{1}{p^2}\right)d_1(\boldsymbol{\phi}) + \frac{1}{p^2}\sum_{k=1}^{W} d_k(\boldsymbol{\phi})k^2.
\end{aligned} \tag{F.2}$$

Now (4.2) yields

$$d_1(\boldsymbol{\phi}) = \sum_{i=1}^{W}\frac{ipq^{i-1}}{1-q^i}\phi_i \tag{F.3}$$

and

$$\begin{aligned}\sum_{k=1}^{W} d_k(\boldsymbol{\phi})k^2 &= \sum_{k=1}^{W}\sum_{i=k}^{W} \frac{\binom{i}{k}p^k q^{i-k}}{1-q^i}\phi_i k^2 \\ &= \sum_{i=1}^{W}\sum_{k=1}^{i} \frac{\binom{i}{k}p^k q^{i-k}}{1-q^i}\phi_i k^2 \\ &= \sum_{i=1}^{W}\left(\sum_{k=1}^{i}\binom{i}{k}p^k q^{i-k}k^2\right)\frac{\phi_i}{1-q^i}.\end{aligned}$$

Using the relation

$$\sum_{k=1}^{i}\binom{i}{k}x^k y^{i-k}k^2 = \begin{cases} x, & i=1, \\ ix(ix+y)(x+y)^{i-2}, & i\geq 2. \end{cases}$$

yields

$$\sum_{k=1}^{W} d_k(\boldsymbol{\phi})k^2 = \sum_{i=1}^{W}\frac{ip(ip+q)\phi_i}{1-q^i}. \tag{F.4}$$

Putting together (F.1), (F.2), and (F.4) yields

$$\text{MSE}(\hat{m}_\phi) \geq \left(\sum_{i=1}^{W}\frac{i(pi+q^{i+1}-2q^i+q)\phi_i}{p(1-q^i)} - m_\phi^2\right)/N \tag{F.5}$$

which concludes the proof. □

Lemma F.2. *Using the observed set sizes* $\mathbb{S} = \{\mathcal{S}_k\}_{k=1}^N$ *the following*

$$\hat{m}_\phi = \frac{\sum_{k=1}^N \mathcal{S}_k}{Np} + \left(1-\frac{1}{p}\right)\frac{\sum_{k=1}^N \mathbf{1}_{\mathcal{S}_k=1}}{N}, \tag{F.6}$$

is an efficient (smallest variance) unbiased estimator of m_ϕ*.*

Proof. We start by noting that

$$m_\phi = [1, ..., W]\phi = [1, ..., W]B^{-1}d(\boldsymbol{\phi}). \tag{F.7}$$

Denote $z = [z_1, \ldots, z_W] = [1, ..., W]B^{-1}$. From Lemma B.1, we have

$$\begin{aligned} z_i &= \sum_{j=1}^{W} j b^\star_{ji} \\ &= \sum_{j=1}^{i} j \binom{i}{j} p^{-i}(-q)^{i-j}(1-q^j) \\ &= (-q/p)^i \sum_{j=1}^{i} j \binom{i}{j} \frac{1-q^j}{(-q)^j} \end{aligned} \tag{F.8}$$

For $i = 1$ (F.8) yields $z_1 = 1$ and for $2 \leq i \leq W$,

$$z_i = (-q/p)^i \left(-\frac{1-q}{q}\right)^i \frac{i}{1-q} = \frac{i}{p}.$$

Therefore,

$$z = \frac{[p, 2, 3, \ldots, W]}{p}.$$

Thus applying the above back into (F.7) yields

$$m_\phi = \frac{m_d}{p} + \left(1 - \frac{1}{p}\right) d_1(\boldsymbol{\phi}), \tag{F.9}$$

where $m_d = \sum_{i=1}^{W} i d_i$ is the expectation of average set size of observed subsets. Rewriting (F.9) using the set sizes $\mathbb{S}$ we get

$$\hat{m}_\phi = \frac{1}{N} \sum_{k=1}^{N} \left(\frac{\mathcal{S}_k}{p} + \left(1 - \frac{1}{p}\right) \mathbf{1}_{\mathcal{S}_k = 1} \right).$$

Based on our assumption that $\{S_k\}_{k=1}^m$ is an i.i.d. sequence, we have that $\{\mathcal{S}_k\}_{k=1}^N$ is also i.i.d. with distribution $d(\boldsymbol{\phi})$. Therefore,

$$E[\hat{m}_\phi] = E\left[\frac{\mathcal{S}_k}{p} + \left(1 - \frac{1}{p}\right)\mathbf{1}_{\mathcal{S}_k=1}\right],$$

and

$$\text{Var}[(\hat{m}_\phi)^2] = \frac{1}{N}\text{Var}\left[\left(\frac{\mathcal{S}_k}{p} + \left(1 - \frac{1}{p}\right)\mathbf{1}_{\mathcal{S}_k=1}\right)^2\right].$$

Since

$$E[\mathcal{S}_k] = m_d = \sum_{i=1}^{W} i d_i(\boldsymbol{\phi}),$$

and

$$E[\mathbf{1}_{\mathcal{S}_k=1}] = d_1(\boldsymbol{\phi}),$$

we have $E[\hat{m}_\phi] = m_\phi$ from (F.9), which indicates that $\hat{m}_\phi$ is unbiased. Then

$$E[(\mathcal{S}_k)^2] = \sum_{i=1}^{W} i^2 d_i(\boldsymbol{\phi}),$$

$$E[(\mathbf{1}_{\mathcal{S}_k=1})^2] = d_1(\boldsymbol{\phi}),$$

and

$$E[\mathcal{S}_k \mathbf{1}_{\mathcal{S}_k=1}] = d_1(\boldsymbol{\phi}),$$

yield

$$\text{Var}[(\hat{m}_\phi)^2] = \frac{\left(1 - \frac{1}{p^2}\right) d_1(\boldsymbol{\phi}) + \frac{1}{p^2}\sum_{k=1}^{W} d_k(\boldsymbol{\phi}) k^2 - m_\phi^2}{N}.$$

From (F.1) and (F.2) we find that $\hat{m}_\phi$ is an unbiased estimator that achieves the Cramér-Rao lower bound (i.e., it is an efficient estimator). □

Lemma F.3. *Let $\hat{m}$ denote an unbiased estimate of the average set size m_θ. Then,*

$$\begin{aligned} \textit{MSE}(\hat{m}_\theta) \;\geq\; & \frac{1}{\eta^2}\Bigg(\sum_{i=1}^{W}\sum_{j=1}^{W}\frac{ij[(J^{(\phi)})^{-1}]_{ji}}{(1-q^j)(1-q^i)} + m_\theta^2\sum_{i=1}^{W}\sum_{j=1}^{W}\frac{[(J^{(\phi)})^{-1}]_{ji}}{(1-q^j)(1-q^i)} - \\ & 2m_\theta\sum_{i=1}^{W}\sum_{j=1}^{W}\frac{j[(J^{(\phi)})^{-1}]_{ji}}{(1-q^i)(1-q^j)}\Bigg). \end{aligned} \tag{F.10}$$

Proof.

$$\begin{aligned} \text{MSE}(\hat{m}_\theta) \;\geq\; & \frac{\nabla M}{\nabla\theta}\left(\frac{\nabla H}{\nabla\phi}(J^{(\phi)})^{-1}\frac{\nabla H}{\nabla\phi}^T\right)\frac{\nabla M}{\nabla\theta}^T \\ =\; & \left(\frac{\nabla M}{\nabla\theta}\frac{\nabla H}{\nabla\phi}\right)(J^{(\phi)})^{-1}\left(\frac{\nabla M}{\nabla\theta}\frac{\nabla H}{\nabla\phi}\right)^T. \end{aligned} \tag{F.11}$$

where $\frac{\nabla M}{\nabla\theta} = (1,\ldots,W)$. Note that

$$\begin{aligned} \left[\frac{\nabla M}{\nabla\theta}\frac{\nabla H}{\nabla\phi}\right]_k &= \sum_{i=1}^{W} ih_{ik} \\ &= \sum_{\substack{i=1\\ i\neq k}}^{W} i\left(-\frac{\theta_i}{\eta(1-q^k)}\right) + k\left(\frac{1-\theta_k}{\eta(1-q^k)}\right) \\ &= \frac{1}{\eta(1-q^k)}\left(k - \sum_{i=1}^{W} i\theta_i\right) \\ &= \frac{k-m_\theta}{\eta(1-q^k)}. \end{aligned} \tag{F.12}$$

Substituting eq. (F.12) in eq. (F.11), we have

$$\begin{aligned} \text{MSE}(\hat{m}_\theta) \;\geq\; & \sum_{i=1}^{W}\sum_{j=1}^{W}\left(\frac{j-m_\theta}{\eta(1-q^j)}\right)[(J^{(\phi)})^{-1}]_{ji}\left(\frac{i-m_\theta}{\eta(1-q^i)}\right) \\ =\; & \frac{1}{\eta^2}\Bigg(\sum_{i=1}^{W}\sum_{j=1}^{W}\frac{ij[(J^{(\phi)})^{-1}]_{ji}}{(1-q^j)(1-q^i)} + m_\theta^2\sum_{i=1}^{W}\sum_{j=1}^{W}\frac{[(J^{(\phi)})^{-1}]_{ji}}{(1-q^j)(1-q^i)} - \\ & 2m_\theta\sum_{i=1}^{W}\sum_{j=1}^{W}\frac{j[(J^{(\phi)})^{-1}]_{ji}}{(1-q^i)(1-q^j)}\Bigg). \end{aligned}$$

□

Similarly to what we did for eq. (B.5), we split eq. (F.10) into three pieces to analyze its behavior.

$$\text{MSE}(\hat{m}_\theta) \geq \frac{1}{\eta^2}\Bigg(\underbrace{\sum_{i=1}^{W}\sum_{j=1}^{W}\frac{ij[(J^{(\boldsymbol{\phi})})^{-1}]_{ji}}{(1-q^j)(1-q^i)}}_{U_1} + \underbrace{m_\theta^2\sum_{i=1}^{W}\sum_{j=1}^{W}\frac{[(J^{(\boldsymbol{\phi})})^{-1}]_{ji}}{(1-q^j)(1-q^i)}}_{U_2} - \underbrace{2m_\theta\sum_{i=1}^{W}\sum_{j=1}^{W}\frac{j[(J^{(\boldsymbol{\phi})})^{-1}]_{ji}}{(1-q^i)(1-q^j)}}_{U_3}\Bigg).$$

Analysis of U_1

$$\begin{aligned}
\sum_{i=1}^{W}\sum_{j=1}^{W}\frac{ij[(J^{(\boldsymbol{\phi})})^{-1}]_{ji}}{(1-q^j)(1-q^i)} &= \sum_{i=1}^{W}\sum_{j=1}^{W}\sum_{k=1}^{W} ij\binom{k}{i}\binom{k}{j}\left(\frac{q}{p}\right)^{2k}(-q)^{-i-j}d_k(\boldsymbol{\phi}) \\
&= \sum_{k=1}^{W}\left(\frac{q}{p}\right)^{2k} d_k(\boldsymbol{\phi})\left(\sum_{i=1}^{k} i\binom{k}{i}(-q)^{-i}\right)^2 \\
&= \sum_{k=1}^{W}\left(\frac{q}{p}\right)^{2k} d_k(\boldsymbol{\phi})\left(\left(-\frac{q}{p}\right)^{-k}\frac{k}{p}\right)^2 \qquad \text{using (G.1)} \\
&= \frac{1}{p^2}\sum_{k=1}^{W} k^2 d_k(\boldsymbol{\phi}) \\
&= \frac{\eta}{p^2}\sum_{i=1}^{W} ip(ip+q)\theta_i \\
&= \eta(\sum_{i=1}^{W} i^2\theta_i + \frac{q}{p}m_\theta).
\end{aligned}$$

Note that U_1 is bounded by the second moment of the distribution θ.

Analysis of U_2

Note that $U_2 = \frac{m_\theta^2}{\theta_i^2} A_2(i)$. Therefore, we conclude that U_2 diverges if either θ_W decreases exponentially in W and $p < a/(a+1)$ or θ_W decreases slower than exponentially in W and $p < 1/2$.

Analysis of U_3

$$\begin{aligned}
\sum_{i=1}^{W}\sum_{j=1}^{W} \frac{j[(J^{(\phi)})^{-1}]_{ji}}{(1-q^i)(1-q^j)} &= \sum_{k=1}^{W} \left(\frac{q}{p}\right)^{2k} d_k(\boldsymbol{\phi}) \sum_{i=1}^{k} \binom{k}{i} (-q)^{-i} \sum_{j=1}^{k} j \binom{k}{j} (-q)^{-j} \\
&= \sum_{k=1}^{W} \left(\frac{q}{p}\right)^{2k} d_k(\boldsymbol{\phi}) \left(\left(-\frac{p}{q}\right)^{k} - 1\right)\left(\left(-\frac{p}{q}\right)^{k} \frac{k}{p}\right) \qquad \text{using (G.2,G.1)} \\
&= \frac{1}{p} \underbrace{\sum_{k=1}^{W} k d_k(\boldsymbol{\phi})}_{\eta p m_\theta} - \frac{1}{p} \underbrace{\sum_{k=1}^{W} \left(-\frac{q}{p}\right)^{k} k d_k(\boldsymbol{\phi})}_{-\eta q \theta_1} \\
&= \eta(m_\theta + \frac{q}{p}\theta_1).
\end{aligned}$$

Thus,

$$U_3 = 2m_\theta \eta (m_\theta + \frac{q}{p}\theta_1).$$

It is interesting to note that, counterintuitively, U_2 goes to infinity for certain values of p and $\boldsymbol{\theta}$ while U_1 and U_3 are always finite, even though the factor $[(J^{(\phi)})^{-1}]_{ji}$ that appears inside the double summation in U_2 is the same factor that appears multiplied by j and ji in U_1 and U_3, respectively.

Proof of Theorem 4.3

Note that U_1, U_2 and U_3 are positive quantities and, moreover, MSE$(\hat{m}_\theta) > 0 \Rightarrow U_1 + U_2 > U_3$. We observe that U_1 diverges if the second moment of θ is infinite, U_2 diverges if $\sum_{j=1}^{W} \left(\frac{q}{p}\right)^j \theta_j \to \infty$ as $W \to \infty$, while U_3 is always finite.

Proof. **1) When θ_W decreases faster than exponentially in W.**

In this case, the second moment of $\boldsymbol{\theta}$ is finite and the sum $\sum_{j=1}^{W} \left(\frac{q}{p}\right)^j \theta_j = O(1)$ for $0 < p < 1$. Therefore, MSE$(m(\mathbb{S})) = O(1)$ for $0 < p < 1$.

2) When θ_W decreases exponentially in W.

The second moment of $\boldsymbol{\theta}$ is still finite. However, we can show that the sum $\sum_{j=1}^{W} \left(\frac{q}{p}\right)^j \theta_j$ is $\Omega(W)$ for $p \leq a/(a+1)$ and $O(1)$ for $p > a/(a+1)$ by using an argument similar to the one used in Section E of Appendix A. Hence, MSE$(m(\mathbb{S})) = \Omega(W)$ for $p \leq a/(a+1)$ and MSE$(m(\mathbb{S})) = O(1)$ for $p > a/(a+1)$.

3) When θ_W decreases more slowly than exponentially in W.

We can show that the sum $\sum_{j=1}^{W} \left(\frac{q}{p}\right)^j \theta_j$ is $\Omega(W)$ for $p < 1/2$ and $O(1)$ for $p \geq 1/2$ by using an argument similar to the one used in Section E of Appendix A. However, the second moment of $\boldsymbol{\theta}$ shows up in U_1 and it can be either finite or infinite. Although it does not affect the bound for $p < 1/2$, in which case we have $\log$ MSE$(m(\mathbb{S})) = \Omega(W)$, it does change the bound for $p \geq 1/2$. In particular, if $p = 1/2$ and $\sum_{j=1}^{W} j^2\theta_j = \omega(1)$, then MSE$(m(\mathbb{S})) = \omega(1)$. On the other hand, if $p = 1/2$ and $\sum_{j=1}^{W} j^2\theta_j \geq O(1)$, then MSE$(m(\mathbb{S})) = \Omega(1)$. Finally, if $p > 1/2$, then MSE$(m(\mathbb{S})) = \Omega(1)$ as well.

□

APPENDIX G

USEFUL IDENTITIES

$$\sum_{j=1}^{k} j\binom{k}{j}(-q)^{-j} = \left(-\frac{q}{p}\right)^{-k}\frac{k}{p} \tag{G.1}$$

$$\sum_{j=1}^{k} \binom{k}{j}(-q)^{-j} = \left(-\frac{q}{p}\right)^{-k} - 1 \tag{G.2}$$

$$\sum_{k=1}^{j} \binom{j}{k}\left(\frac{q}{p}\right)^{k} = \left(\frac{1}{p}\right)^{j} - 1 \tag{G.3}$$

$$\sum_{k=1}^{j} \binom{j}{k}(-1)^{k} = -1 \tag{G.4}$$

$$\sum_{k=0}^{j} \binom{j}{k}(-1)^{k} = \begin{cases} 1 & \text{if } j = 0 \\ 0 & \text{otherwise} \end{cases} \tag{G.5}$$

APPENDIX H

CAN WE LEVERAGE DIVERSITY USING A SINGLE CLASSIFIER?

Intuitively, when a learning model is fitted to the nodes it chose to query, it tends to specialize in one region of the feature space and the search will consequently only explore similar parts of the graph, which can severely undermine its potential to find target nodes.

One potential way to mitigate this overspecialization would be to sample nodes probabilistically, as opposed to deterministically querying the node with the highest score. Clearly, we should not query nodes uniformly at random *all the time*. It turns out that querying nodes uniformly at random *periodically* does not help either, according to the following experiment. We implemented an algorithm for selective harvesting that samples at each step t, with probability p, an uniformly random node from $\mathcal{B}(t)$, and with $1-p$, the best ranked node according to a support vector regression (SVR) model. Table H.1 shows the results for $p = 2.5$, 5.0, 10, 15 and 20%.

0.0%	2.5%	5.0%	10%	15%	20%
760.5 ± 52.1	773.85 ± 34.5	768.0 ± 32.3	770.8 ± 34.1	753.0 ± 59.8	764.7 ± 28.0

Table H.1. Results for SVR w/ uniformly random queries on CiteSeer (at $t = 1500$) averaged over 40 runs. Top line shows probabilty of random query; bottom line shows number of target nodes found.

We observe that the performance does not improve significantly for $p \geq 2.5\%$, either because the diversity is not increasing in a way that translates into performance improvements or because all gains are offset by the samples wasted when querying nodes at random.

Instead of querying uniformly at random, we could query nodes according to a probability distribution that concentrates most of the mass on the top k nodes w.r.t. model scores.

We experimented with several ways of mapping scores to a probability distribution P. In particular, we considered two classes of distributions:

- truncated geometric distribution ($0 < q < 1$):

$$P(v) \propto (1-q)^{\pi(v)-1} q, \quad \text{and}$$

- truncated Zeta distribution ($r \geq 1$):

$$P(v) \propto \pi(v)^{-r},$$

where $\pi(v)$ is the rank of v based on the scores given by the model to $v \in \mathcal{B}(t)$. In each experiment, we set q or r at each step in one of nine ways:

1. Top 10 have $x\%$ of the probability mass; for $x \in \{70, 90, 99\}$.
2. Top 10% nodes have $x\%$ of the probability mass; for $x \in \{90, 99, 99.9\}$.
3. Top $k(t) = \min\{10 \times (1 - t/T), 1\}$ have $x\%$ of the probability mass; for $x \in \{70, 90, 99\}$.

None of the mappings was able to substantially increase the search's performance. In contrast to almost 20% performance improvement seen by SVR under round-robin on CiteSeer at $T = 1500$ (Fig. 5.3), mapping scores to a probability distribution increased the number of targets nodes found by at most 3%.

APPENDIX I

EVALUATION OF MAB ALGORITHMS APPLIED TO SELECTIVE HARVESTING

We experiment with representative algorithms of each of the following bandit classes: Stochastic Bandits – UCB1, Thompson Sampling (TS), ϵ-greedy; Adversarial Bandits – Exp3 [7]; Non-stationary stochastic bandits – Dynamic Thompson Sampling (DTS) [32]; Contextual Bandits – Exp4 [7] and Exp4.P [13]. UCB1 and TS are parameter-free. For ϵ-greedy, Exp3 and Exp4.P we set the probability of uniformly random pulls, to $\epsilon \in \{0.10, 0.20, 0.50\}$, $\gamma \in \{0.10, 0.20, 0.50\}$ and $Kp_{\min} \in \{0.01, 0.05, 0.10, 0.20, 0.50\}$ (respectively). We set parameter γ in Exp4 as $K_p\min$ in Exp4.P. For DTS, we set the cap on the parameter sum $C \in \{5, 10, 20, 50\}$. Interestingly, for each MAB algorithm, there was always one parameter value that outperformed all the others in almost all seven datasets. In Figure I.1 we show three representative plots of the performance comparison between the best parameterizations of each MAB algorithm. Since Exp4 was slightly outperformed by Exp4.P, Exp4 is not shown. These results corroborate our expectations (Section 5.5) that DTS would outperform other bandits in selective harvesting problems.

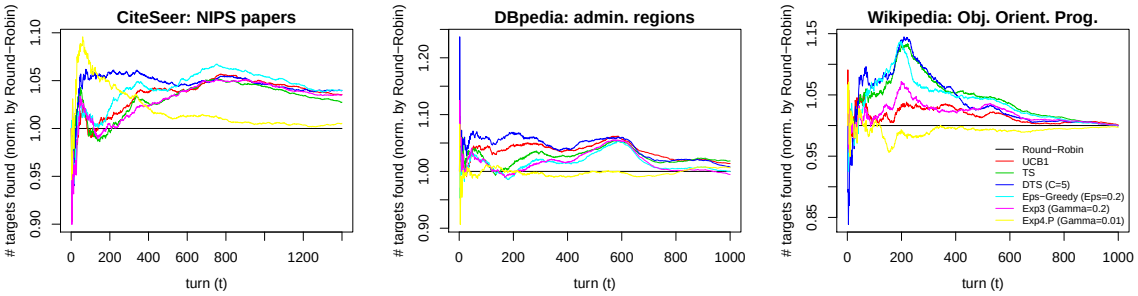


Figure I.1. Comparison between the best parameterizations of each MAB algorithm.

BIBLIOGRAPHY

[1] Achlioptas, D., Clauset, A., Kempe, D., and Moore, C. On the bias of traceroute sampling: Or, power-law degree distributions in regular graphs. *J. ACM 56*, 4 (July 2009), 21:1–21:28.

[2] Achlioptas, Dimitris, Clauset, Aaron, Kempe, David, and Moore, Cristopher. On the bias of traceroute sampling: Or, power-law degree distributions in regular graphs. *J. ACM 56*, 4 (July 2009), 21:1–21:28.

[3] Albert, Réka, Jeong, Hawoong, and Barabási, Albert-László. Attack and error tolerance of complex networks. *Nature 406*, 6794 (2000), 378–382.

[4] Ali, Alnur, Caruana, Rich, and Kapoor, Ashish. Active Learning with Model Selection. *AAAI* (2014), 1673–1679.

[5] Attenberg, J, Melville, P, and Provost, F. Guided feature labeling for budget-sensitive learning under extreme class imbalance. *ICML Workshop on Budgeted Learning* (2010).

[6] Attenberg, Josh, and Provost, Foster. Online active inference and learning. In *KDD* (2011).

[7] Auer, Peter, Cesa-Bianchi, Nicolo, Freund, Yoav, and Schapire, Robert E. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing 32*, 1 (2002), 48–77.

[8] Avrachenkov, K, Basu, P, Neglia, G, and Ribeiro, B. Pay Few, Influence Most: Online Myopic Network Covering. In *IEEE NetSciCom Workshop* (2014).

[9] Avrachenkov, Konstantin, Ribeiro, Bruno, and Towsley, Don. *Improving Random Walk Estimation Accuracy with Uniform Restarts*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 98–109.

[10] Bar-Yossef, Ziv, and Gurevich, Maxim. Random sampling from a search engine's index. *Journal of the ACM 55*, 5 (2008), 1–74.

[11] Baram, Yoram, El-Yaniv, Ran, and Luz, Kobi. Online choice of active learning algorithms. *The Journal of Machine Learning Research 5* (Dec. 2004), 255–291.

[12] Ben-Haim, Z., and Eldar, Y.C. On the constrained Cramér-Rao Bound with a singular Fisher information matrix. *Signal Processing Letters, IEEE 16*, 6 (Jun 2009), 453 –456.

[13] Beygelzimer, Alina, Langford, John, Li, Lihong, Reyzin, Lev, and Schapire, Robert E. Contextual Bandit Algorithms with Supervised Learning Guarantees. *AISTATS* (2011), 19–26.

[14] Bnaya, Z, Puzis, R, Stern, R, and Felner, A. Bandit Algorithms for Social Network Queries. In *SocialCom* (2013).

[15] Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., and Hwang, D.-U. Complex networks: Structure and dynamics. *Physics Reports 424*, 4-5 (2006), 175–308.

[16] Borgs, Christian, Brautbar, Michael, Chayes, Jennifer, Khanna, Sanjeev, and Lucier, Brendan. The Power of Local Information in Social Networks. In *Internet and Network Economics*. Springer Berlin Heidelberg, 2012, pp. 406–419.

[17] Cao, Zhe, Qin, Tao, Liu, Tie-Yan, Tsai, Ming-Feng, and Li, Hang. Learning to rank: from pairwise approach to listwise approach. *ICML* (2007), 129–136.

[18] Chiericetti, Flavio, Dasgupta, Anirban, Kumar, Ravi, Lattanzi, Silvio, and Sarlós, Tamás. On sampling nodes in a network. In *Proceedings of the 25th International Conference on World Wide Web* (Republic and Canton of Geneva, Switzerland, 2016), WWW '16, International World Wide Web Conferences Steering Committee, pp. 471–481.

[19] Dasgupta, Anirban, Kumar, Ravi, and Sivakumar, D. Social sampling. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 2012), KDD '12, ACM, pp. 235–243.

[20] Devroye, Luc. *Non-Uniform Random Variate Generation*, 1 ed. Springer, Apr. 1986.

[21] Duffield, Nick, Lund, Carsten, and Thorup, Mikkel. Estimating flow distributions from sampled flow statistics. *IEEE/ACM Transactions on Networking 13*, 5 (2005), 933–946.

[22] Eagle, Nathan, Pentland, Alex (Sandy), and Lazer, David. Inferring friendship network structure by using mobile phone data. *Proceedings of the National Academy of Sciences 106*, 36 (2009), 15274–15278.

[23] Faloutsos, Michalis, Faloutsos, Petros, and Faloutsos, Christos. On power-law relationships of the internet topology. In *SIGCOMM '99: Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication* (New York, NY, USA, 1999), ACM, pp. 251–262.

[24] Friedman, Jerome, Hastie, Trevor, and Tibshirani, Robert. *The elements of statistical learning*, vol. 1. Springer series in statistics Springer, Berlin, 2009.

[25] Ganti, Ravi, and Gray, Alexander G. UPAL: Unbiased Pool Based Active Learning. *AISTATS* (2012), 422–431.

[26] Ganti, Ravi, and Gray, Alexander G. Building bridges: Viewing active learning from the multi-armed bandit lens. In *UAI* (2013).

[27] Garnett, Roman, Krishnamurthy, Yamuna, Wang, Donghan, Schneider, Jeff, and Mann, Richard. Bayesian optimal active search on graphs. In *MLG* (2011).

[28] Garnett, Roman, Krishnamurthy, Yamuna, Xiong, Xuehan, Mann, Richard, and Schneider, Jeff G. Bayesian optimal active search and surveying. In *ICML* (New York, NY, USA, 2012), ACM, pp. 1239–1246.

[29] Gjoka, Minas, Butts, Carter T., Kurant, Maciej, and Markopoulou, Athina. Walking in facebook: A case study of unbiased sampling of osns. In *Proceedings of IEEE INFOCOM 2010* (March 2010), pp. 1–9.

[30] Gorman, John D., and Hero, Alfred O. Lower bounds for parametric estimation with constraints. *IEEE Transactions on Information Theory 36*, 6 (Nov 1990), 1285–1301.

[31] Gouriten, Georges, Maniu, Silviu, and Senellart, Pierre. Scalable, generic, and adaptive systems for focused crawling. In *Conf. Hypertext Soc. media* (sep 2014), pp. 35–45.

[32] Gupta, Neha, Granmo, Ole-Christoffer, and Agrawala, Ashok K. Thompson Sampling for Dynamic Multi-armed Bandits. *ICMLA* (2011), 484–489.

[33] Heckathorn, Douglas D. Respondent-driven sampling: A new approach to the study of hidden populations. *Social Problems 44*, 2 (1997), 174–199.

[34] Heckathorn, Douglas D. Respondent-driven sampling ii: Deriving valid population estimates from chain-referral samples of hidden populations. *Social Problems 49*, 1 (2002), 11–34.

[35] Helleputte, Thibault. *LiblineaR: Linear Predictive Models Based on the LIBLINEAR C/C++ Library*, 2015. R package version 1.94-2.

[36] Henzinger, Monika R., Heydon, Allan, Mitzenmacher, Michael, and Najork, Marc. On near-uniform url sampling. *Computer Networks 33*, 16 (2000), 295 – 308.

[37] Hohn, Nicolas, and Veitch, Darryl. Inverting sampled traffic. In *IEEE Transactions on Networking* (2006).

[38] Hsu, Wei-Ning, and Lin, Hsuan-Tien. Active Learning by Learning. *AAAI* (2015), 2659–2665.

[39] Hubler, Christian, Kriegel, H-P, Borgwardt, Karsten, and Ghahramani, Zoubin. Metropolis algorithms for representative subgraph sampling. In *2008 Eighth IEEE International Conference on Data Mining* (Dec 2008), pp. 283–292.

[40] Iwamasa, Yuni, and Masuda, Naoki. Networks maximizing the consensus time of voter models. *Physical Review E 90*, 1 (2014), 012816.

[41] Jacquet, Philippe, Mans, Bernard, and Rodolakis, Georgios. Information propagation speed in mobile and delay tolerant networks. *IEEE Transactions on Information Theory 56*, 10 (2010), 5001–5015.

[42] Kay, Steven M. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[43] Khuller, Samir, Purohit, Manish, and Sarpatwar, Kanthi K. Analyzing the optimal neighborhood: Algorithms for budgeted and partial connected dominating set problems. In *SODA* (2014), SIAM, pp. 1702–1713.

[44] Klimt, Bryan, and Yang, Yiming. Introducing the enron corpus. In *First conference on email and anti-spam (CEAS)* (2004).

[45] Kuncheva, Ludmila I. That elusive diversity in classifier ensembles. In *Iberian Conference on Pattern Recognition and Image Analysis* (2003), Springer, pp. 1126–1138.

[46] Kurant, Maciej, Gjoka, Minas, Butts, Carter T., and Markopoulou, Athina. Walking on a graph with a magnifying glass: Stratified sampling via weighted random walks. In *Proceedings of the ACM SIGMETRICS 2011* (New York, NY, USA, 2011), ACM, pp. 281–292.

[47] Kurant, Maciej, Markopoulou, Athina, and Thiran, Patrick. Towards unbiased bfs sampling. *IEEE Journal on Selected Areas in Communications 29*, 9 (September 2011), 1799–1809.

[48] Lakshminarayanan, Balaji, Roy, Daniel M, and Teh, Yee Whye. Mondrian forests: Efficient online random forests. In *Advances in Neural Information Processing Systems* (2014), pp. 3140–3148.

[49] Langford, John, and Zhang, Tong. The Epoch-Greedy Algorithm for Multi-armed Bandits with Side Information. *NIPS* (2007), 817–824.

[50] Lehmann, E. L., and Casella, George. *Theory of Point Estimation*. Springer, 1998.

[51] Lehmann, Erich Leo, Casella, George, and Casella, George. *Theory of point estimation*. Wadsworth & Brooks/Cole Advanced Books & Software, 1991.

[52] Leskovec, Jure, and Faloutsos, Christos. Sampling from large graphs. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 2006), KDD '06, ACM, pp. 631–636.

[53] Leskovec, Jure, and Krevl, Andrej. SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data`, June 2014.

[54] Leskovec, Jure, Lang, Kevin J., Dasgupta, Anirban, and Mahoney, Michael W. Statistical properties of community structure in large social and information networks. In *Proceedings of the 17th International Conference on World Wide Web* (New York, NY, USA, 2008), WWW '08, ACM, pp. 695–704.

[55] Li, Lihong, Chu, Wei, Langford, John, and Schapire, Robert E. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web* (2010), ACM, pp. 661–670.

[56] Liu, Weifeng, Principe, Jose C, and Haykin, Simon. *Kernel adaptive filtering: a comprehensive introduction*, vol. 57. John Wiley & Sons, 2011.

[57] Ma, Yifei, Huang, Tzu-Kuo, and Schneider, Jeff G. Active Search and Bandits on Graphs using Sigma-Optimality. *UAI* (2015), 542–551.

[58] Maiya, Arun S., and Berger-Wolf, Tanya Y. Benefits of bias: Towards better characterization of network sampling. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 2011), KDD '11, ACM, pp. 105–113.

[59] Massoulié, Laurent, Le Merrer, Erwan, Kermarrec, Anne-Marie, and Ganesh, Ayalvadi. Peer counting and sampling in overlay networks: random walk methods. In *Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing* (2006), ACM, pp. 123–132.

[60] McCallum, Andrew, Nigam, Kamal, Rennie, Jason, and Seymore, Kristie. A machine learning approach to building domain-specific search engines. In *IJCAI* (1999), vol. 99, Citeseer, pp. 662–667.

[61] McGregor, Andrew. Graph stream algorithms: a survey. *ACM SIGMOD Record 43*, 1 (2014), 9–20.

[62] Mislove, Alan, Marcon, Massimiliano, Gummadi, Krishna P., Druschel, Peter, and Bhattacharjee, Bobby. Measurement and analysis of online social networks. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement* (New York, NY, USA, 2007), IMC '07, ACM, pp. 29–42.

[63] Móri, Tamás F. The maximum degree of the barabásialbert random tree. *Comb. Probab. Comput. 14*, 3 (May 2005), 339–348.

[64] Murai, Fabricio, Ribeiro, Bruno, Towsley, Donald, and Gile, Krista. Characterizing branching processes from sampled data. In *Proceedings of the 22nd International Conference on World Wide Web* (2013), WWW '13 Companion, International World Wide Web Conferences Steering Committee, pp. 805–812.

[65] Newman, M. E. J. Assortative mixing in networks. *Phys. Rev. Lett. 89* (Oct 2002), 208701.

[66] Newman, Mark EJ. The structure and function of complex networks. *SIAM review 45*, 2 (2003), 167–256.

[67] Olver, Peter. *Applications of Lie groups to differential equations*, 2nd ed. ed. Springer-Verlag, 2000.

[68] Pant, Gautam, and Srinivasan, Padmini. Learning to crawl: Comparing classification schemes. *ACM Trans. Inf. Syst. 23*, 4 (oct 2005), 430–462.

[69] Pfeiffer III, Joseph J, Neville, Jennifer, and Bennett, Paul N. Active sampling of networks. In *MLG* (2012).

[70] Pfeiffer III, Joseph J, Neville, Jennifer, and Bennett, Paul N. Active exploration in networks: Using probabilistic relationships for learning and inference. In *CIKM* (2014).

[71] Rasti, Amir H., Torkjazi, Mojtaba, Rejaie, Reza, Duffield, Nick, Willinger, Walter, and Stutzbach, Daniel. Respondent-driven sampling for characterizing unstructured overlays. In *Proceedings of the IEEE INFOCOM 2009* (April 2009), pp. 2701–2705.

[72] Ribeiro, B., and Towsley, D. On the estimation accuracy of degree distributions from graph sampling. In *51st IEEE Conference on Decision and Control (CDC 2012)* (Dec 2012), pp. 5240–5247.

[73] Ribeiro, Bruno, Gauvin, William, Liu, Benyuan, and Towsley, Don. On myspace account spans and double pareto-like distribution of friends. In *INFOCOM IEEE Conference on Computer Communications Workshops , 2010* (March 2010), pp. 1–6.

[74] Ribeiro, Bruno, Hoang, Minh X, and Singh, Ambuj K. Beyond models: Forecasting complex network processes directly from data. In *Proceedings of the 24th International Conference on World Wide Web* (2015), ACM, pp. 885–895.

[75] Ribeiro, Bruno, Towsley, Don, Ye, Tao, and Bolot, Jean. Fisher information of sampled packets: an application to flow size estimation. In *Proc. of the IMC* (2006), pp. 15–26.

[76] Ribeiro, Bruno, Wang, Pinghui, Murai, Fabricio, and Towsley, Don. Sampling directed graphs with random walks. In *Proceedings of IEEE INFOCOM 2012* (March 2012), pp. 1692–1700.

[77] Ribeiro, Bruno F., and Towsley, Donald F. Estimating and sampling graphs with multidimensional random walks. *CoRR abs/1002.1751* (2010).

[78] Robins, Garry, Pattison, Pip, Kalish, Yuval, and Lusher, Dean. An introduction to exponential random graph (p^*) models for social networks. *Social networks 29*, 2 (2007), 173–191.

[79] Robins, Garry, Snijders, Tom, Wang, Peng, Handcock, Mark, and Pattison, Philippa. Recent developments in exponential random graph (p^*) models for social networks. *Social networks 29*, 2 (2007), 192–215.

[80] Salehi, M., and Rabiee, H. R. A measurement framework for directed networks. *IEEE Journal on Selected Areas in Communications 31*, 6 (June 2013), 1007–1016.

[81] Salganik, Matthew J., and Heckathorn, Douglas D. Sampling and estimation in hidden populations using respondent-driven sampling. *Sociological Methodology 34* (2004), 193–239.

[82] Schein, Andrew I., and Ungar, Lyle H. Active learning for logistic regression: an evaluation. *Machine Learning 68*, 3 (2007), 235–265.

[83] Settles, Burr. Active learning literature survey. *University of Wisconsin, Madison 52*, 55-66 (2010), 11.

[84] Seung, H Sebastian, Opper, Manfred, and Sompolinsky, Haim. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory* (1992), ACM, pp. 287–294.

[85] Stapenhurst, Richard. *Diversity, margins and non-stationary learning*. PhD thesis, University of Manchester, 2012.

[86] Stutzbach, Daniel, Rejaie, Rea, Duffield, Nick, Sen, Subhabrata, and Willinger, Walter. On unbiased sampling for unstructured peer-to-peer networks. *IEEE/ACM Transactions on Networking 17*, 2 (April 2009), 377–390.

[87] Tang, E Ke, Suganthan, Ponnuthurai N, and Yao, Xin. An analysis of diversity measures. *Machine Learning 65*, 1 (2006), 247–271.

[88] Tune, Paul, and Veitch, Darryl. Fisher information in flow size distribution estimation. In *IEEE Transactions on Information Theory* (2011), vol. 57, pp. 7011–7035.

[89] van Trees, Hary L. *Detection, Estimation and Modulation Theory, Part 1*. Wiley, New York, 2001.

[90] Volz, Erik, and Heckathorn, Douglas D. Probability based estimation theory for respondent driven sampling. *Journal of Official Statistics 24*, 1 (03 2008), 79.

[91] Wang, Xuezhi, Garnett, Roman, and Schneider, Jeff. Active search on graphs. In *SIGKDD* (2013), ACM, pp. 731–738.

[92] Whittle, P. Restless Bandits: Activity Allocation in a Changing World. *Journal of Applied Probability 25* (Jan. 1988), 287–298.

[93] Xie, Pengtao, Zhu, Jun, and Xing, Eric. Diversity-promoting bayesian learning of latent variable models. In *Proceedings of the 33st International Conference on Machine Learning (ICML-16)* (2016).

[94] Zhou, Zhuojie, Zhang, Nan, Gong, Zhiguo, and Das, Gautam. Faster random walks by rewiring online social networks on-the-fly. *ACM Trans. Database Syst. 40*, 4 (Jan. 2016), 26:1–26:36.